Forgery Detection by Local Correspondence

Jinhong Katherine Guo

Center for Automation Research
University of Maryland
College Park, MD 20742-3275

Abstract

Signatures may be stylish or unconventional and have many personal characteristics that are challenging to reproduce by anyone other then the original author. For this reason, signatures are used and accepted as proof of authorship or consent on personal checks, credit purchases and legal documents. Currently signatures are verified only informally in many environments, but the rapid development of computer technology has stimulated great interest in research on automated signature verification and forgery detection.

In this thesis, we focus on forgery detection of off-line signatures. Although a great deal of work has been done on off-line signature verification over the past two decades, the field is not as mature as on-line verification. Temporal information used in on-line verification is not available off-line and the subtle details necessary for off-line verification are embedded at the stroke level and are hard to recover robustly. We approach the off-line problem by establishing a local correspondence between a model and a questioned signature. The questioned signature is segmented into consecutive stroke segments that are matched to the stroke segments of the model. The cost of the match is determined by comparing a set of geometric properties of the corresponding sub-strokes and computing a weighted sum of the property value differences. The least invariant features of the least invariant sub-strokes are given the biggest weight, thus emphasizing features that are highly writer-dependent.

Random forgeries are detected when a good correspondence cannot be found, i.e., the process of making the correspondence yields a high cost. Many simple forgeries can also be identified in this way. The threshold for making these decisions is determined by a Gaussian statistical model.

Using the local correspondence between the model and a questioned signature, we perform skilled forgery detection by examining the writer-dependent information embedded at the sub-stroke level and trying to capture unballistic motion and tremor information in each stroke segment, rather than as global statistics. Experiments on random, simple and skilled forgery detection are presented.

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **APR 2000**	2. REPORT TYPE	3. DATES COVERED **00-04-2000 to 00-04-2000**

4. TITLE AND SUBTITLE **Forgery Detection by Local Correspondence**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Language and Media Processing Laboratory,Institute for Advanced Computer Studies,University of Maryland,College Park,MD,20742-3275**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**113**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

Forgery Detection by Local Correspondence

Jinhong Katherine Guo

Forgery Detection by Local Correspondence

Jinhong Katherine Guo

Center for Automation Research
University of Maryland
College Park, MD 20742-3275

Abstract

Signatures may be stylish or unconventional and have many personal characteristics that are challenging to reproduce by anyone other then the original author. For this reason, signatures are used and accepted as proof of authorship or consent on personal checks, credit purchases and legal documents. Currently signatures are verified only informally in many environments, but the rapid development of computer technology has stimulated great interest in research on automated signature verification and forgery detection.

In this thesis, we focus on forgery detection of off-line signatures. Although a great deal of work has been done on off-line signature verification over the past two decades, the field is not as mature as on-line verification. Temporal information used in on-line verification is not available off-line and the subtle details necessary for off-line verification are embedded at the stroke level and are hard to recover robustly. We approach the off-line problem by establishing a local correspondence between a model and a questioned signature. The questioned signature is segmented into consecutive stroke segments that are matched to the stroke segments of the model. The cost of the match is determined by comparing a set of geometric properties of the corresponding sub-strokes and computing a weighted sum of the property value differences. The least invariant features of the least invariant sub-strokes are given the biggest weight, thus emphasizing features that are highly writer-dependent.

Random forgeries are detected when a good correspondence cannot be found, i.e., the process of making the correspondence yields a high cost. Many simple forgeries can also be identified in this way. The threshold for making these decisions is determined by a Gaussian statistical model.

Using the local correspondence between the model and a questioned signature, we perform skilled forgery detection by examining the writer-dependent information embedded at the sub-stroke level and trying to capture unballistic motion and tremor information in each stroke segment, rather than as global statistics. Experiments on random, simple and skilled forgery detection are presented.

Chapter 1

Introduction

1.1 Introduction

Writing is a function of the conscious and subconscious mind and the muscle movements of the body and is thus produced as a stylistic expression of the author. Even though a writer will never produce the exact same signature twice, a signature has many personal characteristics. For this reason, it is used and accepted as proof of authorship or consent on personal checks, credit purchases and legal documents. Although signatures are commonly used to authorize major transactions, they are typically verified informally, if at all, and only on rare occasions when a document is questioned, does verification involve forensic experts. With the rapid development of computer technology, however, the task of signature verification has stimulated interest beyond the realm of forensic science.

1.2 Automatic Signature Verification

A person's style of writing begins to develop when the person learns to write. The style may change over the years, and may vary with circumstances (type of writing instrument, type of surface being written on, and environmental factors); but in spite of all these variations, a person's writing is remarkably consistent. A person's signature may have special features—it may be stylish or unconventional; but there is great consistency among the signatures of a given person.

Signatures have many individual characteristics, and signature verification requires analysis of the signature based on features which are believed to be invariant to the writer. Because these features are closely linked to the writer, signatures are often used and accepted as proof of authorship or consent for financial and legal documents. Unfortunately, in today's society little or no attempt is made to verify signatures, manually or otherwise, unless a discrepancy occurs.

As an increasing number of transactions, especially financial, are being authorized via signature, methods of automatic signature verification must be developed if their authenticity is to be checked on a more regular basis. For example, signatures on personal checks are seldomly checked carefully on site. Only when a customer raises doubt, a transaction is revisited and the signature is examined. As another example, when we use credit cards for purchases, perhaps 50% of the time a sales person turns the card over and checks the signature. The fast checkout machines that are gaining popularity at supermarkets nowadays do the processing by a sweep of the card, and the cashiers do not even bother to look at the card. Only when a transaction is in dispute, the signature on the sales slip is revisited. These market trends of reducing checkout time are welcomed by consumers. However, transaction fraud costs industry and retailers

thousands of dollars each year. If automatic verification can be achieved, we can still keep up with the fast pace of financial transactions, yet prevent unnecessary losses.

Signature verification attempts to confirm that a given signature is genuine, or equivalently, to identify a questioned signature as a forgery. Forgeries are typically classified into three types: random, simple and skilled. Random forgeries occur when the author simply uses an invalid signature, possibly his/her own, without using knowledge of the writer's name or style of writing. These are the simplest to detect because they differ globally from the genuine signature. Simple forgeries involve using the writer's name without knowledge of the writer's style. Skilled forgeries involve attempting to mimic the style of the writer and can be difficult to detect, even for experts. Skilled forgeries are often sub-classified into traced and simulated forgeries.

Automatic signature verification has been the subject of considerable research for over 25 years. Both on-line and off-line verification have been studied; this chapter deals with the off-line case, which is more difficult because information about pen movement is not available. A review of the literature on signature verification through 1993 can be found in [1] and [2]. A comprehensive survey on on-line and off-line handwriting recognition can be found in [3].

As we will see in Chapter 2, most of the past approaches to signature verification fall into two categories according to the method of acquisition of the data: on-line and off-line. On-line data records the motion of the stylus while the signature is produced, and includes location, and possibly velocity, acceleration and pressure, as functions of time. Off-line data is an image of the signature. Although on-line verification is of great interest for "point of sale" applications, we will concentrate primarily on off-line verification which is of interest to check clearing houses, tax processing centers and other locations where a hard copy of the signature is obtained.

1.2.1 The problem

Handwriting recognition, including characters, numbers and signatures, has been an important research area for many years. Optical character recognition (OCR) techniques have developed to the point that both graphics and text can be recognized reasonably well in machine-generated documents and have led to the development of commercial OCR packages. Recognition of handwritten symbols and signatures, however, still remains a difficult problem. Handwritten symbols are often connected, thus making it difficult to segment words into symbols and symbols into strokes. The variation between different writers, or even variations in multiple instances obtained from the same writer, make symbols difficult to recognize. In this sense, OCR is a classification problem which attempts to perform a many-to-one mapping of all variations of a given symbol into a single class, the symbol "a" for example.

Signature verification deals with a somewhat different problem. In general, signature verification tries to verify the signature based on features which are known to be invariant to the writer and which can thus be used to show that a given signature did or did not come from a given writer. Although we must still accept intra-class variability, we must now exclude inter-class similarity. Signature verification has many of the difficulties encountered in the domain of handwritten OCR, and the variations which must be used to discriminate are often very subtle. Many of the features which make a signature unique are embedded at the stroke level. The strokes, however, may be even harder to segment because signatures are usually highly stylistic and often do not follow traditional writing style. Many factors can affect signatures and other handwriting, including injuries, illness, age and emotional state, as well as external factors such as alcohol and drugs. Speed of writing, position of the writer, writing instrument and temperature can also play roles in the variation, thus blurring the class boundaries [4].

Document examiners consider all these factors when they analyze a questioned document and often use a unique set of instance-specific features to prove or disprove authorship.

As we will see in Chapter 2, most of the past research on automatic signature verification has made use of global geometric properties of the signature. Such properties are usually sufficient to distinguish one person's signature from another's; thus *random forgeries*, in which the forger does not sign the correct name, are relatively easy to detect. It is somewhat more difficult to detect *simple forgeries*, in which the forger writes the correct name but makes no attempt to imitate the genuine author's handwriting. Such forgeries become easier to detect if a genuine signature and a forged signature can be compared letter by letter or stroke by stroke, because the shapes of the corresponding strokes will differ. However, such local comparison is difficult to do automatically, because it is hard to segment a signature consistently into strokes. *Skilled forgeries*, in which the genuine author's signature is simulated or traced, are still harder to detect; but because such forgeries are usually written slowly, they can be distinguished from genuine signatures (which are written rapidly and smoothly) by local analysis at the substroke level to detect pressure variations.

1.3 Signatures and Forgeries

1.3.1 Handwriting

In an on-line signature, the trajectory of the pen movement can be approximately modeled by representing the pen tip coordinates as functions of time:

$$x = x(t)$$

$$y = y(t)$$

Throughout this thesis, we will refer to establishing a local correspondence between a model and a questioned signature. In later chapters we will often refer to the pen trajectory, in various representations, as the model. For completeness, in this section we describe a general model for a signature as a dynamic event that produces a static image, and use it as a basis for examining features for discrimination.

The Pen Trajectory

An off-line signature image consists of more than just the trajectory of the pen tip. The thickness of the signature in the image depends on the size of the pen tip as well as other factors determined by the angle at which the pen is held, the force exerted on the pen and the type of pen that is being used.

When the pen is perpendicular to the paper and the force on the pen is evenly distributed on the tip, the pen tip area in which ink is deposited should be a circle, as shown in Figure 1.1(a). The radius of this circle can be represented by an expression of the form

$$R = r + \delta(t)$$

where the constant r is the radius of the pen tip and $\delta(t)$ is a "marginal" factor that depends on the paper and how the ink is dispensed, and is a function of the force f and angle α of the pen at time t:

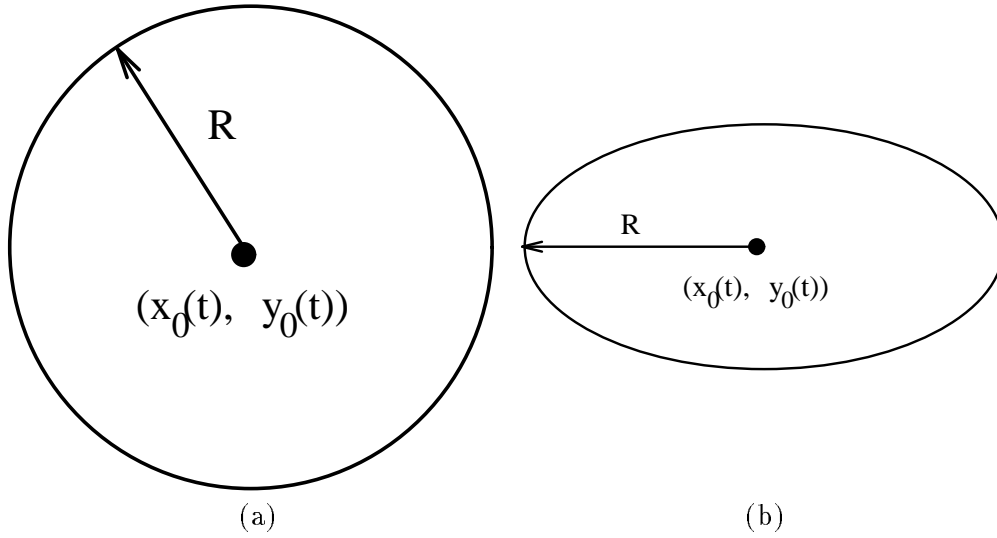$$\delta(t) = F_1(\alpha, f, t)$$

Figure 1.1: The ink deposited by a pen (a) when the pen is perpendicular to the paper; (b) when the pen is not perpendicular to the paper.

In general, the pen is not perpendicular to the paper, and the pen tip area is therefore not a circle. Since this area is small, we will approximately represent it as an ellipse, as shown in Figure 1.1(b).

The darkness $D(x, y)$ at a point in the pen tip area depends on the force and angle of the pen, as well as on the distance between the point and the center (x_0, y_0) of the pen tip:

$$D(x, y) = F_2(\alpha, f, t, \sqrt{(x - x_0)^2 + (y - y_0)^2})$$

The image grey level $G(x, y)$ is then proportional to the sum of $D(x, y)$ for all (x_0, y_0) on the pen tip trajectory:

$$G(x, y) = \sum D(x, y) = F_3$$

As the angle of the pen and the force on the pen change over time, $D(x, y)$ changes along the trajectory of the signature. This rhythmic change, which gives rise to the line quality of the writing, is what we need to capture in order to examine a questioned signature.

The Cross-Section of the Pen Trajectory

To have a better understanding of the local properties embedded at the stroke level, we need to consider the darkness along the cross-sections of the pen trajectory. We can represent this darkness using polar coordinates (ρ, ϕ) centered at the position $x_0(t), y_0(t)$ of the pen tip center at a given time, as illustrated in Figure 1.2.
Note that

$$\rho = \sqrt{(x - x_0)^2 + (y - y_0)^2}$$

The darkness D in the ideal case, when the pen is perpendicular to the paper, should be symmetric about (x_0, y_0), i.e. should depend only on ρ but not on ϕ, and the darkness of the cross-section of a stroke should be a symmetric function of ρ, as shown in Figure 1.3.
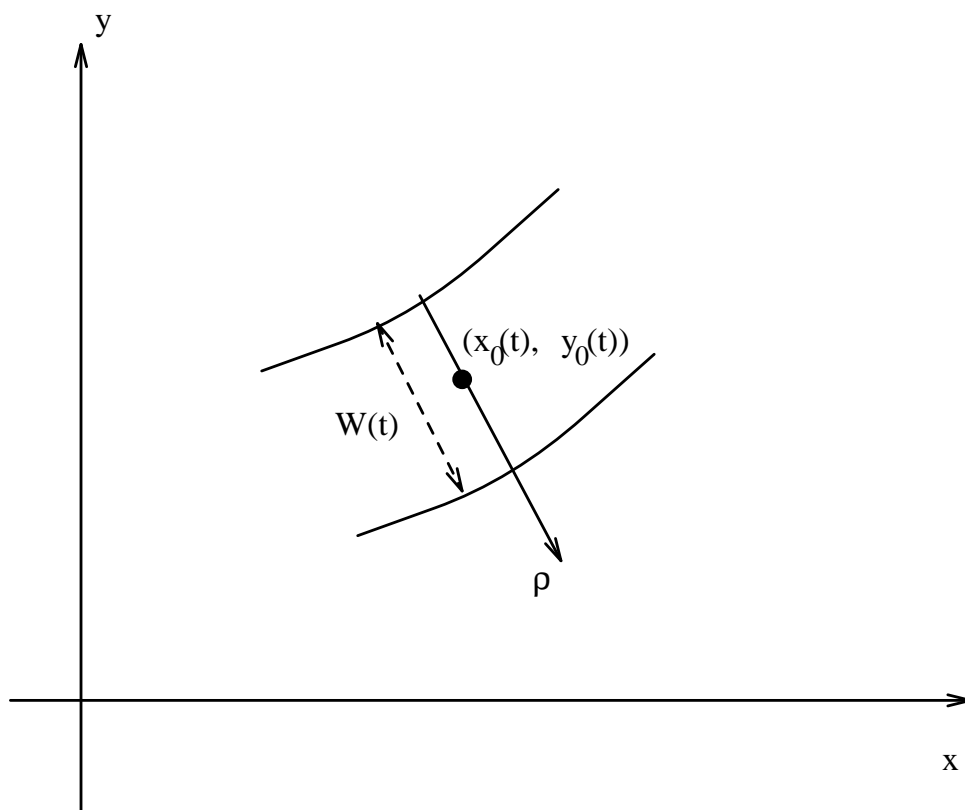
y

$(x_0(t),\ y_0(t))$

W(t)

ρ

x

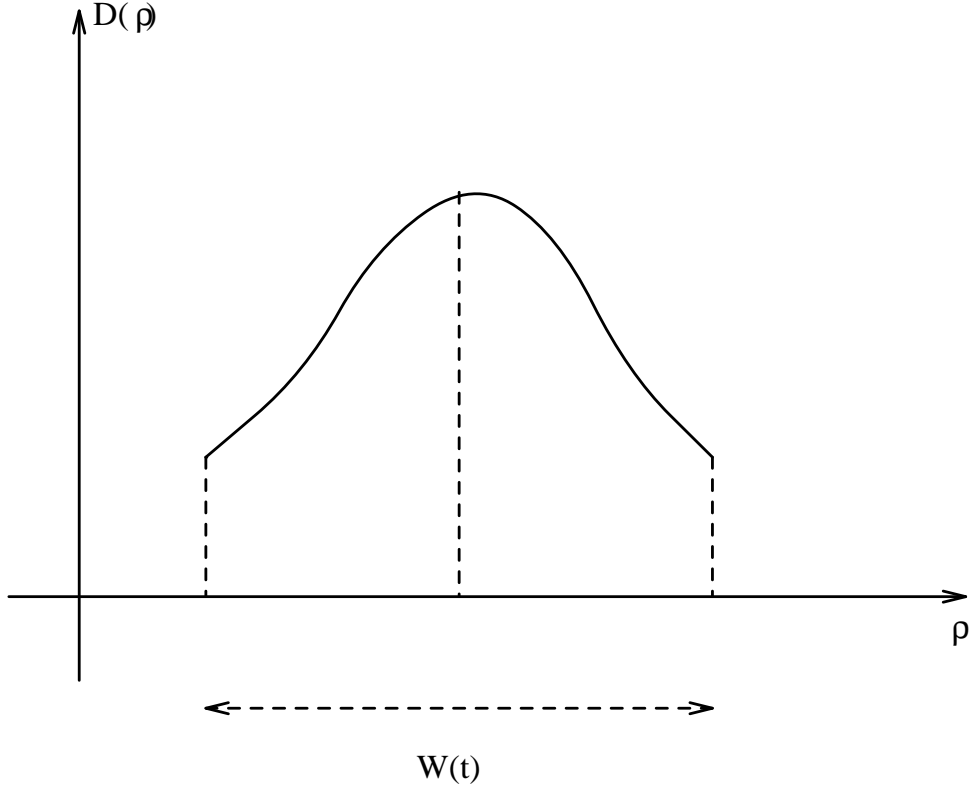Figure 1.2: A cross-section of the pen trajectory.

Figure 1.3: Darkness distribution in a stroke cross-section in the ideal case.

When the pen is not perpendicular to the paper, the darkness distribution is no longer symmetric. However, we can approximate this distribution by an asymmetrically windowed Gaussian, as shown in Figure 1.4, since the width is small.
Thus we can model the darkness distribution in a cross-section by

$$D(\rho(t)) = \frac{1}{\sqrt{2\pi}\sigma(t)} \exp\{-\frac{(\rho(t) - \mu(t))^2}{2\sigma^2(t)}\}W(t)$$

where

$$W(t) = w + \delta(t)$$

$$\sigma(t) = F_4(\alpha, f, t)$$

$$\mu(t) = F_5(\alpha, f, t)$$

$$\delta(t) = F_6(\alpha, f, t)$$

are all functions of the angle α and the force f at time t.
When people sign their names, the angle and force on the pen change with time as the pen makes turns, loops, and upward and downward strokes. From the above discussion, we see that the darkness cross-section, and thus the gray level $G(x, y)$, changes along the stroke. From Figure 1.4 we observe that the gradient magnitude along both edges of the stroke changes as well. We have not attempted to specify the functions F_i in detail here, but our discussion provides a qualitative basis for examining local features of a signature for skilled forgery detection, which we will discuss in detail in Chapter 5.
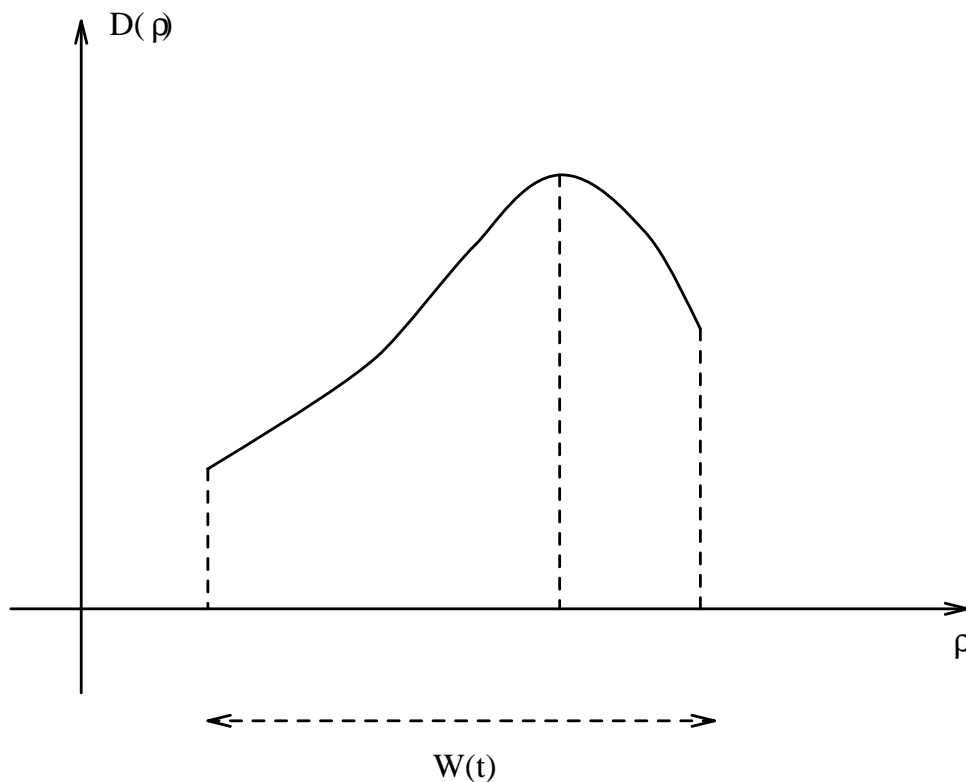
Figure 1.4: Darkness distribution in a stroke cross-section in the general case.

1.3.2 The nature of genuine signatures

Although various factors such as health, age and alcohol can affect a person's writing [5][6], writing style is developed when a person learns to write and develops certain consistencies, according to expert document examiners [7]. Adolescence is the transitional time for handwriting maturity [8]. After that period, handwriting has stabilized considerably [9]. When considering the uniqueness of one's signature, rhythm is a factor that is hard to imitate. Rhythm is a harmonious recurrence of stress or impulse or motion. Harmony of action is based mainly on the circle or the ellipse, and is a succession of connected curved motions rather than a succession of straight movements and sudden stops. The genuine writer has his or her style of varying pen pressure which produces clear-cut accentuated strokes which increase and decrease in width, especially at the beginnings and endings, to vanishing points. A free, uninterrupted motion naturally makes a smooth, uniform, continuous curved or straight line, while a change of direction, stop, tremor, or unevenness in an intermediate part of what ordinarily is a single stroke indicates hesitation or a changing movement impulse. Artistic quality, another factor to be considered, is the proportion of strokes similar to each other in width which are affected by shading and unconscious emphasis. Since writing instruments have changed, shading is not as characteristic as it was years ago, but changes of width and force along the stroke are still significant.

A number of movements employed during writing are characteristic and are hard to imitate from a layout or model or from knowledge about the writer. These movements include finger, hand, forearm and whole-arm movement and their combinations and gradations [7]. Finger movement generates irregular connections between letters and parts of letters and is usually slow

7

and labored and lacks rhythm. It is the movement nearly always employed in forged writing. Hand movement produces very narrow and angular connections at the tops and bottoms of letters. The alignment of writing of this kind is very uneven and many words have an upward tendency. Forearm movement generates smooth, clear-cut, symmetrical and rapid strokes and is the style of writing that is forged with greatest difficulty because the method by which it is produced is the farthest removed from the slow motion employed in carefully following a copy.

The size of a person's signature can be different under various conditions. For example, when signing a form, the signature may be confined to a box. However, a genuine signature shows fixed proportions of the heights of letters to the width of the signature. Slant is another characteristic that is typical for each person.

In analyzing a signature, the first and last strokes of words and separate capital letters are significant parts. Writers sign their names with confidence at the first stroke. The last stroke of the signature is not important to the writer so he or she does it ballistically and unconsciously. The design and artistic quality of capital letters are also very hard to imitate. People may write M, N, T, F, S, A, C, D with different numbers of strokes and different additional touches [10].

1.3.3 The nature of forgeries

As previously mentioned, forged signatures fall into three categories: random, simple and skilled forgeries. Random and simple forgeries are the easiest to detect. Skilled forgeries require a model signature to either imitate or trace. The imitation may be produced in careful, studied detail, but the signature produced often lacks variation in pen pressure and has blunt ending strokes. A forger makes more sudden stops in order to carefully study the genuine signature, while the genuine writer is more confident in writing and signs with more freedom. The forger may not necessarily produce a disconnected line, but often produces a line with angularity and changes of direction. If the skilled forger is signing from memory, the forgery may reveal some of the writing style of the forger since he or she may not remember all the details. Thus, the forgery may contain improper letter designs and the movement usually is slow, laborious and abnormal. It also tends to be letter by letter, stroke by stroke and lacking in rhythm. Since the lines or strokes show the quality of speed and continuity of motion and the degree of muscular skill, forgeries often have poor line quality [11]. The edges of the stroke show the angle between the writing instrument and the stroke. Even in the case of a traced forgery, it would be hard for the forger to simulate the angle of the pen that the genuine writer would use. A forger would hesitate at the first stroke, making it with tremor or a change of direction. He or she also treats the last stroke, which is unimportant to the writer, with self-consciousness. The general features a document examiner would look at are:

- Slant of upward and downward strokes

- Spacing of letters

- Size of capital and small letters

- Proportions of individual letters and between letters

- Connections of letters and connections between capital and small letters

- Capitals, especially those mentioned in the last section

- Quality of line

- Position of pen

- Alignment

A document examiner would also be looking for inconsistent letter formations, and would check some special features such as erasures, retouching, "t" crossings and "i" dots, etc. Leung et al. [12][13] did some statistical research on traced and simulated forgeries.

1.3.4   The problem of signature verification

Given a questioned signature, we need to make a decision as to whether it is a genuine signature or a forgery. To do this, we extract features $F_1, F_2, \ldots, F_N$ from the questioned signature. This N-tuple of features $\mathcal{R} = [F_1, F_2, \ldots, F_N]$ is our observation.

We have two hypotheses:

$$H_0 : \text{forged}$$

$$H_1 : \text{genuine}$$

Given $\mathcal{R}$, we want to decide, with minimum error, whether the questioned signature belongs to H_1 or H_0. Specifically, we want to minimize the total error, i.e.,

$$\min\{p(H_1|H_0) + p(H_0|H_1)\}$$

where we have assumed that the a priori probabilities $p(H_1)$ and $p(H_2)$ are equal.

1.4   Proposed Research

In this research we will address the problems associated with off-line signature verification. We will explore methods of extracting and representing features used for discrimination and use the features to build a model for each signature. We will use the model to guide the analysis of questioned signatures and attempt to identify random, simple and traced or simulated forgeries.

Research has progressed more rapidly in applications using on-line data, in part because discriminating information involved in writing, such as changes of speed and force, are available on-line but are lost in off-line data. It has been known that this is the information that is hard for forgers to imitate. For many applications, however, using off-line data is unavoidable. One difficulty in off-line signature verification lies in the fact that it's hard to segment the signature into components which are meaningful stylistically. For this reason, few approaches attempt to extract specific types of features, but rather use more global characterizations. We propose a model-driven approach to signature verification which extracts key stylistic features from signatures to be used for discrimination. We assume that the available model or reference signature contains some amount of temporal information. The model can be based on on-line data gathered a priori, or extracted [14] or traced [15] from a reference signature, or simply traced manually with a digitizer. In practice, some banks let their clients sign using a digitizer when they open an account. This would be a source of on-line data. A traced model will still be helpful in segmenting a signature image and classifying the segments into upward and downward or retraced strokes, etc.

1.4.1 Detection of simple and random forgeries

Random forgery occurs when the forger signs another name, and is often overlooked when there are massive numbers of documents to be processed. Simple forgery occurs when the forger signs the correct name in his or her own style. A majority of forgeries are simple forgeries, since forgers try to take advantage of the massive processing load in credit purchases and check cashing. In our proposed system, one goal of establishing a correspondence is to eliminate random forgeries.

Since the system is model-based, the model can drive the process to establish a correspondence with the questioned signature image. Critical points are extracted from the model data. The strokes are segmented at critical points. We can adopt some ideas from on-line signature segmentation [16] and use them for segmenting the traced model. Doermann [14] and Lee and Pan [15] did some work on recovering temporal information of the strokes in signature images. Since we know what we are looking for from the model, we can also incorporate some of our heuristic knowledge about the model into the tracing of the signature image.

After segmenting the stroke segments, we can use an elastic matching technique [17] [18] to make a correspondence between the model and the data. In order to match the data against the model, a cost function (distance metric) has to be designed. The cost function for each segment $d(i, j)$ is a weighted sum of differences of feature values between the model and the questioned signature. The weights are determined from a training set of known genuine signatures. The overall distance between the questioned signature and the model is

$$D = \min_{|m(i)|} \sum_{i=1}^{N} d(i, m(i))$$

where N is the length of the data and $m(i)$ is a warping function mapping the index i to the index of the model.

1.4.2 Detection of skilled forgeries

After random forgeries are taken out of consideration, simulated or traced forgeries need more detailed and careful study. Simple forgeries will also be subject to verification at this stage if they have not been eliminated by the correspondence procedure. In Section 1.3, we discussed the characteristics of both signatures and forgeries. In our system, we want to look at features as document examiners would, as closely as possible. The features considered by our system are:

- The gradient magnitude on the edges of the signature. This gradient is different in an original signature from that in a forgery and corresponds to the position of the pen.

- The gradient direction on the edges of the signature. This corresponds to the slant and smoothness of the writing.

- The change of gray level along the signature. This is related to the change of force while writing. This measurement should be different for the ballistic writing of the writer and the nonballistic writing of the forger. This also corresponds to line quality.

- Stroke width along the signature. This corresponds to the ballisticness of the writing.

Skilled forgeries can be hard to distinguish from genuine signatures. In a traced forgery, the structure features, such as the slant of the signature and the connections between stroke segments, can be the same as in a genuine signature. But the forgery lacks rhythm and has

(a)   (b)

Figure 1.5: An example of a signature and its traced forgery: (a) signature (b) forgery.

poor line quality. Traced forgeries show even more hesitation than simulated forgeries since tracing is drawing rather than writing. Figure 1.5 shows an example of a genuine signature and its traced forgery. The signature shows the writer's ballistic style while the forgery shows more hesitations and unnecessary stops along the stroke. We can also observe from Figure 1.5 that the signature ends in a natural vanishing ending at the upper right conner. The forgery has a more bulb-like ending. Along the long strokes, the forgery shows more tremor, especially on the vertical stroke of the "d". So, the edges of the stroke are not as smooth as those in the signature. Sometimes the forgery also has careful retouchings to repair the mistakes made during tracing. We will discuss how to quantitatively measure these features.

1.5 Structure of the Dissertation

In this thesis, we address the problem of signature verification, both for random forgery detection and for skilled forgery detection. In Chapter 2 we review previous work on the subject. In Chapter 3 and Chapter 4, we address the problem of random forgery detection by making a local correspondence between the model and questioned signature. The difference between the two chapters is in the method used to make the correspondence. In Chapter 3, we introduce an edge-based segmentation algorithm for corresponding the model stroke segments with the stroke segments of the questioned signature, whereas in Chapter 4 we use a model-based segmentation algorithm. In Chapter 5 we describe a method of skilled forgery detection. Finally, Chapter 6 concludes this thesis.

Chapter 2

Previous Work

In this chapter we summarize work that has already been done on both random or simple forgery detection and skilled forgery detection.

2.1 Random and Simple Forgery Detection

A great deal of work has been done in the area of off-line signature verification over the past two decades [1] [2]. Unlike on-line verification, the prospect of producing a commercial off-line verification system is not yet close to realization; nevertheless, steady progress is being made. One of the main themes of our approach is detailed analysis of the questioned signature at the local stroke level, which implies a need for a correspondence between the questioned signature and an example of (or model for) the genuine signature. In this literature review, we focus primarily on work that is relevant to the distinction between approaches which consider primarily features of the signature as a whole, and approaches which make use of local context.

Previous work on off-line signature verification has dealt primarily with random forgeries, where the signatures may not be the same name, and simple forgeries, where stylistic differences are prevalent. Many researchers therefore found it sufficient to consider only global features of the signature.

Such work included the use of Fourier coefficients or moments [19] to identify random forgeries. Nemcek and Lin [20] used 14 features extracted from the Hadamard spectrum and a maximum likelihood classifier. For simple forgeries, the error rates were 11% type I (false dismissals of genuine signatures) and 41% type II (false acceptances of forgeries). Chuang [21] used global features extracted from the upper, middle and lower zones of the signatures. The features related to each zone and their ratios were compared using weighted distance and the results were around 20% for both type I and type II errors. Recent work using global features was done by Zhang, et al. [22] who used a vector quantization technique. Feature extraction was based on down-sampling subregions of the image. A "neural gas" model was used for vector quantization. The error rate was around 90% in the experiment. Mizukami, et al. [23] used an extracted displacement function and a similiarity measure. The best result was a 24.9% error rate.

As signature databases become larger and researchers move toward more difficult simple and skilled forgery detection tasks, we see a progression not only to more elaborate classifiers, but to increased use of local features and matching approaches.

As early as the 1970s, Nagel and Rosenfeld [24] used an approach based on the positions of ascenders and descenders for simple forgery detection assuming the spelling of the signature is known. Their approach uses a weighted distance involving both global and local features. Globally, it uses the ratio of the width of the signature to the height of the short and long

strokes. Locally, it considers both the ratio of the heights of the tall letters to the heights of the short letters that follow them immediately, and the slopes of the tall letters. With 0% type II error, the type I error was 8%.

Many early systems which did make use of local features used statistics such as averages or histograms of these local features to represent the signatures.

In the 1980s, Ammar [25] calculated the statistics of dark, high-pressure pixels to identify changes in the global flow of the writing. The work was aimed at skilled forgery detection, where geometric information about the letters is far less informative. Ammar's high-pressure regions are actually the dark pixels in the signature image that correspond to high-pressure points on the writing trace. His main feature is the ratio of the number of dark pixels to the total number of pixels in the signature. Even though the dark pixels themselves are localized, the features extracted are global. This was also one of the first attempts to consider dynamic information in the static image for verification.

Stroke orientation has also been used for verification. Sabourin [26] applied a classifier to statistics of the orientations of the gradient vectors at each pixel of the signature. The work is based on the assumption that the stroke orientation is stable from one signature to the next. A histogram of the local angular information of the intensity gradient is extracted as a feature vector. The features then were classified by a maximum-likelihood classifier and a linear discriminant nearest-neighbor classifier. With the maximum likelihood classifier, the type I error was 1.5% and the type II error was 7%.

Other work of the group headed by Sabourin is based on shape features derived from the directional PDF (Probability Density Function)[27]. This work was targeted at random forgeries, as most forgeries are random forgeries. A neural network was introduced for verification and training. Later attempts to improve the results focused on using neural networks for classification [28, 29]. Other approaches based on neural networks can be found in [30, 31, 32, 33, 34].

Realizing that the directional PDF does not have knowledge about the locations of local measures, Sabourin et al. [35] [36][37] developed and used an extended shadow code as a feature vector to incorporate both local and global information into the verification process. Extended shadow codes are local projections of the signature on a bar mask consisting of horizontal, vertical and diagonal bars. It is possible, therefore, to bypass the segmentation process by projection on the bar mask, but without a segmentation process, it is not possible to obtain a more accurate comparison of the local features. These codes are not invariant to scale, translation or rotation, although they allow for some tolerance.

The Sabourin group has also incorporated a fuzzy technique to add some pseudo-dynamic information (such as pen-up and pen-down events) to the extended shadow code [38], and has used a pattern spectrum [39] derived from successive applications of morphological operators in rectangular areas called retinas. These approaches based on the shadow code and shape spectrum, however, assume that all signatures are of similar size, and that the corresponding strokes of the signatures fall into approximately the same regions of the signature. Shape descriptors can be applied only to those rectangular retinas that are of perceptual interest [40], but this too requires invariance of the sizes and locations of corresponding strokes. They have also tried to incorporate more local features into the verification system by using similarity of shape matrices [41]. Even though shape matrices can be used as a representation of planar shapes, they do not provide for local pairwise stroke comparisons. The average of the type I and type II errors using the minimum distance classifier ranged between 0.848% and 1.664%.

Nouboud and Plamondon [42] used curve comparison algorithms to compare the outline curves of a reference and questioned signature. Although the outline curve reflects global shape, it also incorporates some local information. The curve is obtained by scanning the image for

the first white-to-black and last black-to-white transitions of each line. The minimum distance from the questioned signature to each reference signature is computed and compared with a threshold. With 0% type II error, the type I error was 15.2%.

Lee and Lizarraga [43] applied a classifier to the histograms of properties extracted using morphological operators with structuring elements representing different slopes. The detection rate was approximately 90%.

Since global properties and even statistics of local features alone have only limited discrimination power, it was also recognized during the same period that geometrical and topological methods may contribute to the state of the art. Pavlidis et al. [44][45] developed an active deformable model for approximating the external shape of a signature. This model originated from snake theory. Their algorithm needs to have a clean signature image as input. On 180 signature images, in one paper, the performance was 78.89% correct detection, 18.33% inconclusive and 2.78% false detection; in the other paper, it was 70.83% correct detection, 13.33% inconclusive and 15.84% false detection.

Han and Sethi [46] also used a set of geometrical and topological features. The geometrical features included horizontal bars, vertical bars and loops, and the topological features included end points, branch points, crossing points, convex points and concave points. These features were mapped into two strings which were used for indexing into the signature database. The indexing scheme was based on string hashing. The best result was 97.5% correct detection.

In relevant research of writter identification, Wilensky [47], et al, presented a new metric for invariant image recognition to recognize and characterize handwritten words. By defining a distortion insensitive distance, they were able to compare a test image with the prototype image. Experiments on signature classification were presented with error rate of 96%. Nearest neighbor classification scheme was used and a match is said to occur when one of the top 9 matches is the genuine writer.

Many of these approaches incorporate both local and global information. However, they involve no direct correspondence between a model and a questioned signature. It is clear from the literature that as we move toward skilled signature verification, experts rely on very detailed stroke-level differences to identify forgeries [7]. In our earlier work [48] we presented an algorithm for making such a correspondence. A disadvantage of this algorithm was that it was based on an edge image computed from the grey level signature image; thus the performance of the algorithm depended on the quality of the edge detection. Furthermore, the process of segmenting the questioned signature was not model-based. In our more recent work, we present a model-based segmentation algorithm which is directly applied to grey level images of questioned signatures without preprocessing. We will show how our algorithm can be used to identify random forgeries by matching the questioned signature to the model of the genuine signature.

Since our work makes use of a pseudo-on-line model to which we match the questioned signature, it is also appropriate to mention some of the on-line techniques. The major difference between on-line and off-line data is that information like pen position and velocity can easily be recorded in on-line environments. While exact velocity information is not recoverable in off-line data, information such as the pen position as a function of time is embedded at the stroke level, and can in principle be recovered by segmenting the signature into stylistically meaningful segments. Doermann [14] used stroke features to recovery dynamic information.

Pan and Lee [15] applied a set of heuristic rules to thinned signature images. By tracing the thinned signature following junction point rules, starting and ending point rules, and backtracking and top-bottom-left-right rules, the pen movement that generates the signature can be recovered. Special rules for letters such as "a" "d" "g" and "q" that involve counterclockwise

circular pen motion must also be used. It should be pointed out that since signatures are highly stylish and cursive, thinning artifacts can be more severe.

Unlike off-line signature verification, the features used for on-line verification are always those with dynamic properties [49][50]. Since on-line signals contain a significant amount of data, some researchers have focused on data acquisition. Earlier on-line work made use of global profiles of features such as speed and pressure [51] [50]. By segmenting the signatures, local comparison is possible [52]. Some segmentation techniques are model-guided [53]. Other techniques such as elastic matching [54] or time warping [55][56] have also been widely used, as have regional correlation [57] and tree matching. There are also approaches based on methods such as Hidden Markov Models [58][59][60][61]. The choice of method is based primarily on processing time and on the features and parameters needed for each technique.

Another aspect of on-line signature verification which led to accelerated progress is that of data acquisition. With the development of digital pens/tablets, more information could be recorded more efficiently. For example, a digital pen can measure acceleration [49], or can measure the angle of the pen and the force exerted on it [62]. As an alternative to the traditional pen/tablet approach, Munich and Perona used a camera-based system [63][64] to capture on-line signatures.

Since we use dynamic programming for matching the questioned signature with the model, it is also worth mentioning work involving dynamic programming. Nouboud and Plamondon [42] used dynamic programming for curve comparison, where the curve was obtained from the outline of the signature. They used Paliwal's method [65]. Yoshimura et al. [66] applied a dynamic programming technique to the projection profile of the frequency of black pixels on the x-axis for Japanese signature verification. The average error rate was around 12.9%. These algorithms apply dynamic programming or time warping primarily to a function or curve. In our approach, since we obtain a segmentwise correspondence before the matching process, traditional elastic matching techniques for on-line handwritten character recognition [18] [17] can be applied at the segment level.

2.2   Skilled Forgery Detection

Skilled forgery detection is a challenging task even for expert document examiners. The differences between a genuine signature and a skillfully forged one can be subtle. No two signatures are exactly the same. The task of detecting a skilled forgery has to distinguish between the variations among genuine signatures and the true differences between a signature and a forgery. In the case of automatic signature verification, noise and degradation of the image due to the scanning and digitization process make this even harder. Unlike random forgery detection, not much work has been done on skilled forgeries in the automatic signature verification field. Most of the published work has been done by Ammar and his group.

As early as the 80's, Ammar et al. [25] did work on detection of skilled forgeries. He calculated the statistics of dark (therefore, presumably high-pressure) pixels and used them to identify changes in the global flow of the writing. His main feature was the ratio of the number of dark pixels to the total number of pixels in the signature. This was also one of the first attempts to extract "dynamic" information from a static image for signature verification. In his experiment, the forgeries were written using pencils.

Later work of Ammar [67] was based on reference patterns, namely the horizontal and vertical projections of the signature image. Averaging the horizontal and vertical projections over the training sample yields vertical and horizontal reference patterns. The projection of the questioned signature and the reference are compared using Euclidean distance. Using the

leave-one-out method, the best result achieved was a 10.25% error rate. In [68], Fang, et al, used the reference patterns together with a smoothness measure of the contour of the signatures on skilled forgery detection. The experiment showed an average of 21.7% of both type I and type II errors which is an 1.1% improvement, comparing with 22.8% average errors when using the reference patterns alone on the same data set.

The work we present in this thesis is different from the research discussed above in that we focus on information embedded on the substroke level. Instead of obtaining global statistics of the dark pixels or horizontal and vertical information, we try to measure properties that are writer-dependent. The statistics of dark pixels are used to capture uniformity in the forging process, but they may be more attributable to the writing instrument, and may be more significant for pencils than for other modern writing instruments such as ball-point pens. The horizontal and vertical projections are primarily suitable for measuring differences between the general slant and shape of the model and questioned signatures. It may be difficult to tell the difference between the two if the forger is familiar with the writer's style and skilled at simulating the general appearance of the signature.

Chapter 3

Edge Based Local Correspondence for Detecting Random Forgeries

3.1 Introduction

We approach the off-line signature verification problem by establishing a local correspondence between a model, derived from one or more genuine signatures, and a questioned signature. Random forgeries which do not confirm to the model can be eliminated. By making the correspondence, local features can be further examined down to the stroke segment level, providing information that can be used for simple forgery detection. The correspondence will also be used for detection of skilled forgeries, simulated or traced.

In Section 3.2 we provide an overview of the problem of off-line signature verification and a description of our approach. We discuss model acquisition in Section 3.3, and explain the process of segmenting the questioned signature in Section 3.4. In Section 3.5 we describe how we make the correspondence between the model and the data. Our experimental results are presented in Section 3.6, and possible extensions are discussed in Section 3.7.

3.2 Background

3.2.1 The problem

As a person learns to write the person's writing style develops consistencies. Expert document examiners try to exploit these consistencies and identify ones which are both stable and difficult to imitate. From this knowledge, the expert can examine a reference signature and a questioned signature, establish a correspondence between them and look for differences at the stroke level.

The movements employed in writing are characteristic and hard to imitate. Different movements generate different stroke styles and connections. Thus, a person's signature is characterized by a sequence of stroke segments of specific types that tends to be consistent for a given writer. The connections between consecutive stroke segments and the styles of the curves tend to remain consistent. It makes sense to try to exploit such properties for verification.

When experts consider a reference signature, they see not only global features, but also details of the pen movements that generate the signature. Even non-experts have an impressive ability to retrieve both spatial and temporal information from the reference. When examining a questioned signature, document examiners usually ignore the feature differences between the reference and the questioned signature that are less significant. They can immediately focus on (or weight) the features that are consistent for the writer and examine these features in the reference signature. The component missing in the current research on off-line signature verification is the ability to examine local features of the reference signature. Most current
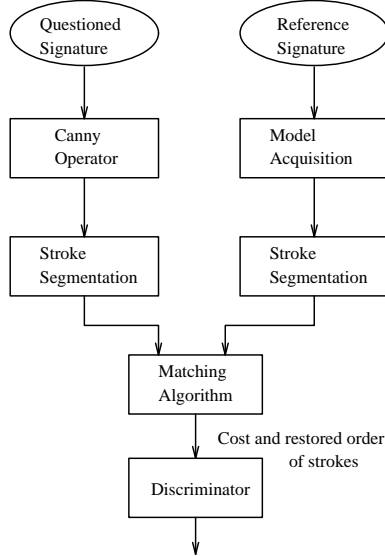
17

Figure 3.1: The scheme of our approach.

algorithms are based on either global features of the entire signature or global statistics of the local features.

3.2.2 Our approach

In this chapter, we present an algorithm that takes a fundamental step in establishing a local correspondence between the model and the questioned signature. The algorithm examines features extracted from a set of training signatures written by each writer. Since we want to access the signature at the stroke level, our first step is to segment the questioned signature into consecutive stroke segments, consistent with the process used to create the known signature. This involves determining the correct ordering of the strokes, including retraced strokes. In many cases, signatures do not follow traditional writing style, making them hard to segment. It is also necessary to have a model that reflects both static and dynamic (pen movement) information. From the reference signature, we obtain a model which contains information about the stroke segments and their order, as well as information about the consistency of some of the structural features. The model includes a *tracing* of the signature, either gathered a priori on-line, extracted automatically [14], or traced manually with a digitizer, and a weight matrix whose elements are associated with each feature of each stroke segment. We use the model to establish a local correspondence between the reference and questioned signatures. Figure 3.1 summarizes the overall system. A discussion of the model acquisition process is presented in Section 3.3.

When processing the questioned signature a similar segmentation into constituent strokes is required. Fortunately, the ordering is not essential since it can be refined during matching. It seems natural to consider thinned images in analyzing handwriting, and in general a "correct" thin line provides a compact representation of the writing. However, artifacts have always been a problem in thinning algorithms especially when dealing with writing samples that overlap or cross. For example, when a loop is collapsed so that it touches itself, thinning is not able to recover it. In our approach, edge gradient information is used to obtain edge pixels along candidate strokes. We trace the edge image during the segmentation process, in cooperation

18

Figure 3.2: Example of size and location variations within a person's signature.

with the model, and recover the ordering of the stroke segments and retraced strokes. An edge image does not typically have the artifacts that occur in a thinned image, so, for example, it can be used to extract loops even if they are collapsed since the outside contour is typically stable. Our questioned signature processing algorithms are described in Section 3.4.

After we have extracted a representation for the questioned signature a matching process attempts to establish a correspondence between the model and the questioned signature, as described in Section 3.5.

In our approach, we do not constrain the image size to the size of the model, nor do we assume regularity in the stroke locations. As we can see in Figure 3.2, even though one may be able to normalize a set of signatures to have similar sizes, the extents of the letters can vary from one situation to another. Five signatures of the same person are overlaid in the Figure. The first strokes all start at about the same place, but the space between the first name and last name can be different, making the signatures of different lengths. If some of the signatures have a middle initial, while others do not, the difference between the locations of the stroke segments becomes even greater. The space each letter spans, especially the first letter in the first or last name, also varies more than the letters in the middle, making the letter locations different from signature to signature.

When we use our algorithm, random forgeries are easily identified when a suitable correspondence cannot be found. Simple forgeries can be identified when a correspondence can be found but the variation of the questioned signature from the model (or the cost of matching) is high. Identification of skilled forgeries, especially traced forgeries, requires further analysis of both local and global properties of the signature after the correspondence has been completed; it will be covered in a later chapter.

3.3   Model Acquisition

In most previous work on off-line signature verification, models are images of known signatures and are used to extract spatial features for comparison with these same features extracted from questioned signatures. Our model, on the other hand, contains both spatial and relative temporal information about the writing process as a sequence of consecutive points that record the movement of the writing instrument in generating the signature, along with pen-up and pen-down events. Our model also contains personal information about the writer, reflected by weights assigned to the different features of each stroke segment. Every person has a personal style in different parts of his/her signature; the way he/she treats different strokes may be different. For example, a person may not care about the size of a certain signature segment, but the curvature of that segment may always be consistent. In this case, we need to give more weight to curvature than to size. The weights on the different features are estimated using a database of signature images of signatures called reference signatures. We will discuss the

features we extract and the procedure for computing the weights in Sections 3.3.2 and 3.3.3. The temporal aspect of the model can be obtained either by using a tablet to acquire reference signatures on-line, or by using an input device to trace hard copies. Since this manual effort, and the training to estimate the weights on the features, need be done only once, it is not a significant bottleneck for the system.

3.3.1 Acquisition of pen movement information for constructing a model

In this section, we discuss the procedure of obtaining the pen movement information of the model. Most of the signatures we used are from an existing signature database. The pen movements for these signatures were recovered by tracing a hard copy of one of the signatures.

Although a traced model may not always correctly represent the original writing order of the signature, the signatures to be verified will still have a consistent relationship with the traced model. For example, a person may write a certain letter in an unconventional order, but we trace the hard copy image in the conventional way since we do not know the stroke order. Our matching process will force the questioned signature to correspond to the traced model. Even though the model is not necessarily in the right order, all the questioned signatures will be compared to the model consistently.

After obtaining the model, we segment it into stroke segments at its junction and end points. Thus, the segmented model contains stroke-level information, including the stroke segments, the traces of the pen movements along the strokes, and even information about retraced strokes (which is generally lost if we simply segment an image of the signature).

We generate one on-line or traced representation for each person. Even when there are minor variations, our algorithm can adapt. For example, the second letter "u" in Figure 3.3 may or may not intersect with the letter "L" before it. If the "u" in our model does not intersect while the "u" in the questioned signature does, the second portion of the upward segment of the "u" will match the upward segment in the model, leaving the first portion to match to nothing, or vice versa. The effect of this on the cost function (see Section 3.5) will not be great enough to reject this signature as a simple forgery. Since the most likely stroke is matched to the one in the model (in this case, the upward segment above the last segment of the "L"), the effect on examining the local stroke features to detect skilled forgeries will also be small. Our algorithm would work if the questioned signature had a middle initial while the model did not, or vice versa, since we allow the middle initial to match to nothing. These examples show that if there are not too many variants among a person's signatures, we can generate one model which reflects all the variants and incorporates the most likely situation. In rare cases, where there are major variants, multiple models must be acquired, one for each different style. Ideally, a composite model could be generated as a synthesis of multiple examples.

Figure 3.3 shows an example of the part of a model that reflects pen movement, without connections or explicit ordering information. The ordering information is embedded in the model by specifying a set of consecutive points. To make the model complete, information about stroke segment features and the degree of consistency of those features is needed. These are described in the next two sections.

3.3.2 Feature extraction

In this and the next section, we discuss the features we extract from the model and how we obtain the weights that we give to the different features of different stroke segments. The features we use include the average curvature of the stroke segment, its incoming and outgoing
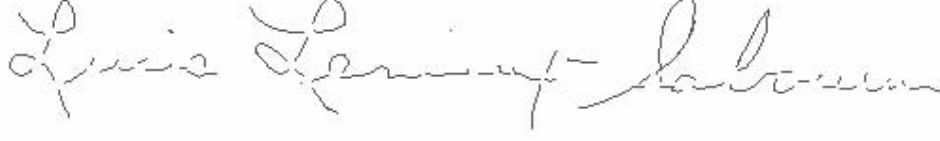
Figure 3.3: An example of the aspect of a model that reflects pen movement.

angles, its relative position and its relative size.

Average Curvature

The curvature feature that we use is the average curvature of the stroke segment. Average curvature is a useful feature for matching purposes because each person writes curves and loops differently. In the process of stroke segment correspondence, we want to match curves to curves and loops to loops, but we want high cost in a situation where the two ends of a stroke segment are similar to those in the model while the shape of the segment is very different. The curvature at each point of the segment is computed using the re-sampling method [69]. The formula is as follows:

$$\kappa(i) = \frac{\theta(i) * G'_\sigma}{1.107}$$
$$\theta(i) = \tan^{-1}\left(\frac{y_{\text{res}}(i+1) - y_{\text{res}}(i)}{x_{\text{res}}(i+1) - x_{\text{res}}(i)}\right)$$

where $\kappa(i)$ is the curvature at point i; $x_{\text{res}}, y_{\text{res}}$ are the coordinates of the re-sampled points on the curve; and G'_σ is the Gaussian differential filter:

$$G_\sigma(i) = \left(\frac{1}{\sigma\sqrt{2\pi}e^{-i^2/2\sigma^2}}\right)_{i=-m,m}, \quad m = \lceil k\sigma \rceil$$

We then compute

$$d_{\text{avgc}} = |C_{\text{sig}} - C_{\text{mod}}|$$

where C_{sig} and C_{mod} are the average curvatures of the stroke segment in the questioned signature and the model respectively. In training to estimate weights for the model, we compute curvature differences between signatures in the reference signature set and the model.

The curvature feature may give misleading information under certain circumstances. Since we are using signed rather than absolute curvature, the average curvature of an "s" can be the same as that of a line. However, curvature is not our only feature, and the other features may compensate for this. Thus, in the "s" example, the incoming and outgoing angle features should compensate. This also illustrates why we need to assign different weights to different features of different segments.

Initial/Final Angles

This feature is defined by

$$d_{\text{inang}} = |S_{\text{inang}} - M_{\text{inang}}|$$

21

$$d_{\text{outang}} = |S_{\text{outang}} - M_{\text{outang}}|$$

where S_{inang} and M_{inang} are the incoming angles of the segments in the questioned signature and model respectively, while S_{outang} and M_{outang} are the outgoing angles of the segments in the questioned signature and model respectively.

The incoming angle is the angle perpendicular to the gradient direction at the starting point of the segment. Thus it reflects the direction to the last segment. The outgoing angle is the angle perpendicular to the gradient direction at the last point of the segment; it reflects the direction to the next segment, or the starting direction of the next segment. Thus, these directions reflect the connections between consecutive stroke segments.

For robustness, the incoming angle is computed as the slope of a line fit to the first few points of the segment, while the outgoing angle is computed as the slope of a line fit to the last few points of the segment. There has been some concern in the literature over the use of direction information because of the unreliability of the information extracted from the last stroke segment of the signature; for instance, the last segment may be cut off when there is not enough space in the document, so that the last outgoing angle may not provide reliable information. However, the connections between letters, and the connections between stroke segments, tend to be reliable for a given writer. These angles are important features in segmentation as well as in signature matching.

Position

This feature is defined by

$$d_{\text{pos}} = |S_{\text{pos}} - M_{\text{pos}}|$$

where S_{pos} and M_{pos} are the x- or y-positions of the segments in the questioned signature and the model respectively.

The position is calculated as the x- or y-distance from the left or upper side of the bounding box of the signature to the center of the bounding box of the segment, divided (respectively) by the x- or y-extent of the signature's bounding box. Figure 3.4 illustrates the definition of the position features. The position of a segment is an important property, since the other properties alone may match a segment of the model to the wrong segment of the data. For example, the word "data" has two a's; it is possible that a segment in the first "a" matches a segment in the second "a". The size of a signature may vary from time to time, but the relative positions of the strokes are likely to be consistent.

Size

The size difference property is defined by

$$d_{\text{size}} = |S_{\text{size}} - M_{\text{size}}|$$

where S_{size} and M_{size} are the arc lengths (computed by joining successive sample points with straight line segments) of the segments of the questioned signature and model respectively. The size of a segment is normalized by the length of the diagonal of the bounding box of the signature. An example is shown in Figure 3.5.

The sizes of signatures can be different for many reasons. The environment of the writing, the type of document, and even the scanning process can affect the size. But the size of a segment relative to the size of the signature is likely to be relatively invariant. For example, if a person writes the letter "L" in his/her name with a long starting stroke, he/she usually does
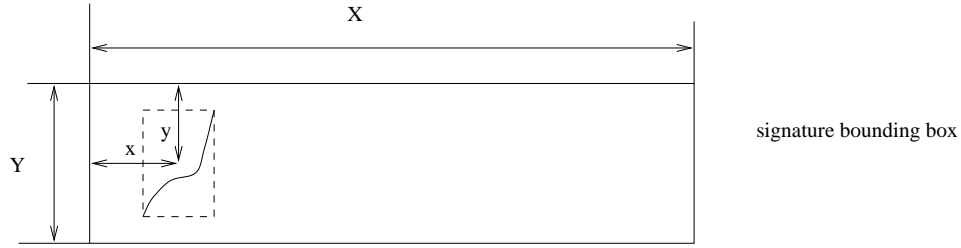
$$S_{xpos} = \frac{x}{X} \qquad\qquad S_{ypos} = \frac{y}{Y}$$

Figure 3.4: Illustration of x and y positions.
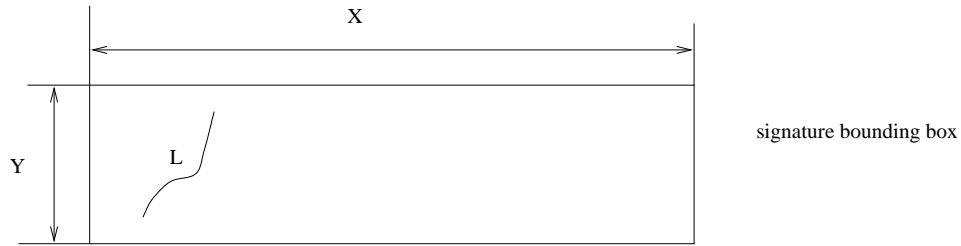
L: arc length

$$S_{size} = \frac{L}{\sqrt{X^2 + Y^2}}$$

Figure 3.5: Illustration of size.

it consistently. Even though there may be changes in absolute size, the proportions of a stroke relative to the other strokes, especially the one immediately following it, are usually consistent and are useful properties in comparing segments.

3.3.3 Training

As discussed in the above sections, some features of a stroke segment may not be as important as others. For example, the outgoing angle of the last stroke segment in the signature may not be important since the boundary of the page may have caused the last stroke to be cut short. As another example, for a very small segment, the incoming and outgoing angles are not very meaningful because of the nature of the algorithm used to calculate the angles. We would like to emphasize the invariants of a particular writer's signature, and not attempt to define general match measures for all writers.

Suppose we have a training set S_{train} of N genuine signatures. For each signature S_j in S_{train}, we compute the differences between the feature values of that signature and the model. We refer to this difference as "cost".

23

Let $\text{Cost}_j^k(i)$ be the cost of property k of the i th segment of signature S_j. We compute the standard deviation of $\text{Cost}_j^k(i)$ over $j = 1, \ldots, N$ and denote it by $\text{Std}^k(i)$:

$$\text{Std}_k(i) = \frac{1}{N} \sqrt{\sum_{j=1}^{N} (\text{Cost}_j^k(i) - \frac{1}{N} \sum_{j=1}^{N} \text{Cost}_j^k(i))^2}$$

The smaller $\text{Std}_k(i)$ is, the more consistent property k is for that writer on segment i. We then let

$$w_k(i) = \frac{1}{\text{Std}_k(i)}$$

where $w_k(i)$ is the weight given to property k of segment i. By choosing the weights in this way, we put more emphasis on the properties that are more consistent in that specific segment for the given writer. This is analogous to what a document examiner would do when examing a questioned signature, i.e., try to identify properties that are especially consistent for the given writer.

The $w_k(i)$'s are computed using a training set and stored as part of the model. We now describe the algorithm for making a correspondence between the model and a questioned signature.

3.4   Questioned Signature Processing

In order to segment a questioned signature into strokes and establish its correspondence with the model, preprocessing of the signature image is necessary. Since we have only a single, static instance of a signature image, care must be taken to accurately segment this image.

As mentioned earlier, we use edge information as a basis for extracting strokes, as opposed to using a thinned representation. There have been improvements in thinning algorithms to combat the artifacts caused by thinning, but accurate thinning of handwritten characters, especially signatures, remains a challenge. Although thinning works well for isolated strokes, collapsed loops and crossings at sufficiently acute angles tend to lose their structure when thinned. Figure 3.6(b) shows examples of the thinning artifacts in a signature. The edge image, on the other hand, tends to preserve the information necessary to recover loops and crossings, at least in part, since at least one contour is typically visible.

In our approach, we apply edge detection to the grey level image, so as not to lose information through binarization. The Canny edge detector [70] was selected because of its ability to detect edges in low-contrast areas. A drawback of using the Canny operator, however, is that the edges are not necessarily continuous. To overcome this, we apply a merging process to the results of the initial segmentation; we will discuss this process in Section 3.4.1. First, we will describe the initial segmentation into simple stroke segments and collapsed loops.

3.4.1   Stroke segmentation

Figure 3.6(c) shows an example of a signature and its edge image obtained via the Canny operator. The edge image preserves the spatial stroke information well and can be used to reliably extract strokes. In addition to the edge image, the Canny operator also generates gradient information for each edge pixel.
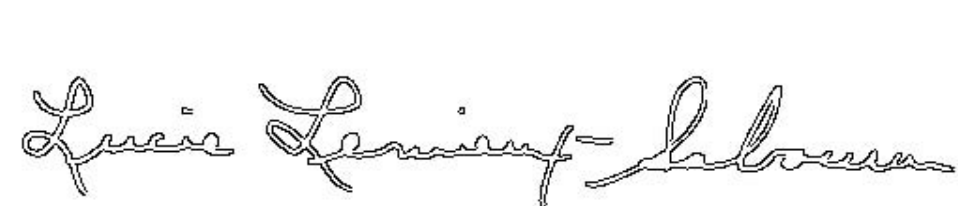
The stroke segmentation process is summarized in Figure 3.7. Strokes are segmented at junction (crossing) and end points. We classify the segments as simple strokes and collapsed loops. Simple strokes are stroke segments that do not interfere with others and appear to be

24
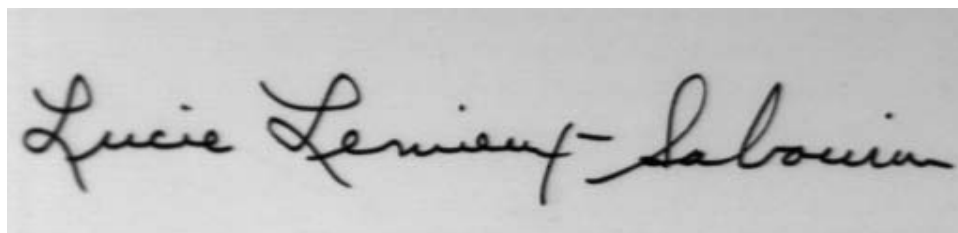
(a)
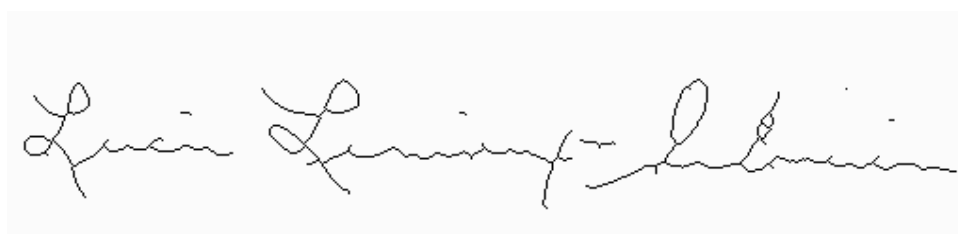
(b)

(c)

Figure 3.6: (a) A signature sample scanned at 300 dpi; (b) the thinned image of (a); (c) the edge image of (a).
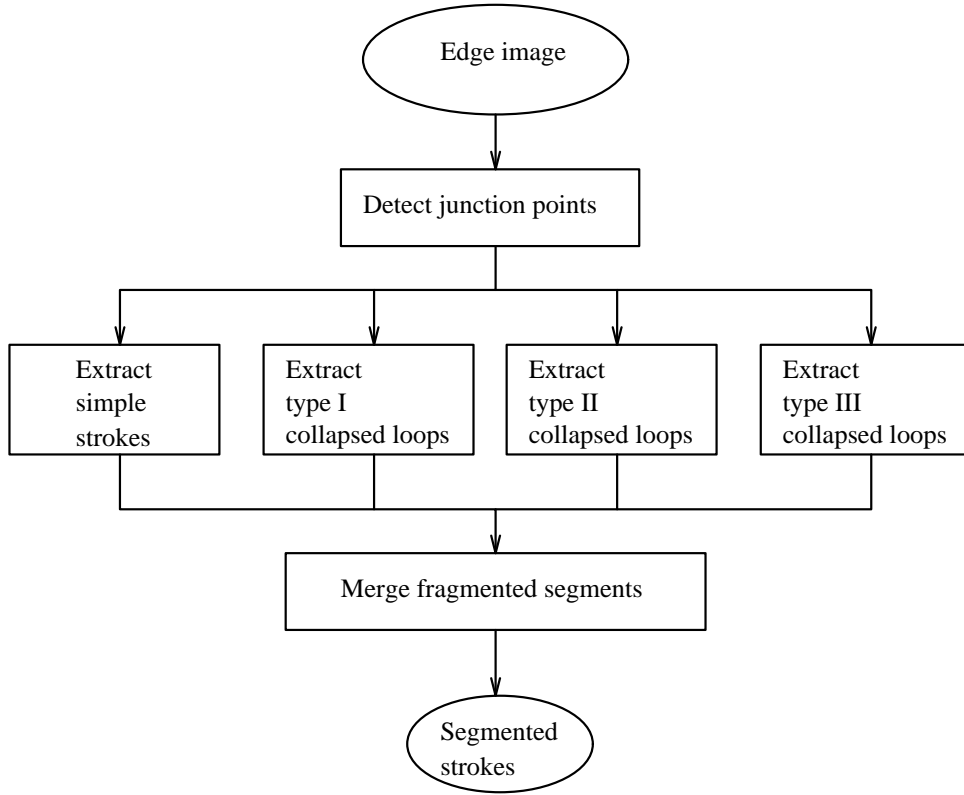
Figure 3.7: Stroke segmentation.

elongated. Well-kept loops and retraced strokes are also considered to be simple strokes, since both of their edges are detectable. In collapsed loops, the inner edges of the strokes are too close to each other, so that one of them is not visible in the edge image. Such loops are handled differently in our algorithm, where a thinned version will be generated from the edge image.

Extracting Simple Stroke Segments

To extract simple stroke segments, we attempt to link edge pixels across the segment by scanning in the direction opposite to the gradient at each edge pixel obtained from the Canny operator. Figure 3.8 shows an example of the detection of a segment of a simple stroke. As we trace along the edges, we find the gradient direction at each pixel, such as P_1, and scan across the stroke to a point P_2 on the opposite edge of the same stroke that corresponds to P_1. The distance between P_1 and P_2 is the width of the stroke at P_1(or P_2). If the width is less than a threshold T, which is chosen to just exceed 80% of the stroke widths in the signature, we generate a simple stroke hypothesis and determine the stroke's midline. The midpoint $((P_1.x + P_2.x)/2, (P_1.y + P_2.y)/2)$ is taken to be a point of the stroke segment.

At junction points, the gradient direction changes more sharply than on normal, relatively straight strokes, and the tracing of the segment stops. The stroke segment is taken to be the sequence of midpoints between the two assumed junction points (its starting and ending points). The tracing of another stroke segment is initiated at the next edge point.
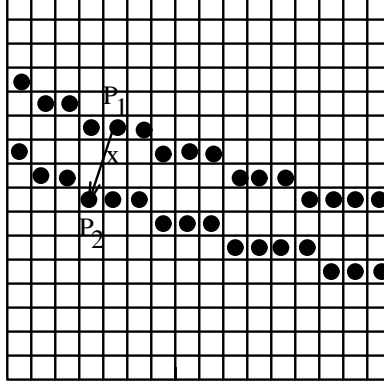
Figure 3.8: Illustration of finding the stroke segments of a simple stroke.

Extracting Collapsed Loops

If P_1 is on the outer edge of a collapsed loop, P_2 may fall on the other side of the outer edge of the loop since the inner edges may not be detectable. Such a case can be detected because the width will be larger than the threshold T, derived from other strokes in the signature. For example, the upper part of the "e" in Figure 3.6 is a collapsed loop where the hole has disappeared because two strokes have merged into one. By using the edge information, we are able to decide that it is a collapsed loop and process it accordingly. When we classify a stroke segment as a collapsed loop, we take its outer contour to be the stroke segment.

Collapsed loops can generally be classified into one of three types. Type I is a long merged stroke such as the loop in an "l" or "e". Type II is a loop connected with another letter, like an "a" or "o" in the middle of a signature. The third type is an isolated loop like a single "o". These three types of loops are shown in Figure 3.9.

Figure 3.9(a) shows an example taken from the "e" in the middle section of the signature in Figure 3.6. As we can see from Figure 3.9(a), the loop has collapsed into one thicker segment. As we trace to P_1, we find the corresponding point P_2 on the other side of the edge. Since we have assumed that the distance between P_1 and P_2 is the width of the segment at that point, we find that the width is greater than the threshold T. If P_1 and P_2 are on the same collapsed loop, all the widths at the points between P_1 and P_2 on the same contour should be greater than the threshold T. We check whether P_1 and P_2 are on the same loop by checking four equally spaced points between P_1 and P_2 on the edge. If the widths between P_1 and P_2 are greater than T and P_1 and P_2 are on the same loop, we conclude that we have the first type of collapsed loop. The outer contours of the loop between P_1 and P_2 are taken as the stroke segments, shown as the thicker curves in Figure 3.9(a).

The second type of collapsed loop is shown in Figure 3.9(b). This example is taken from the letter "o" in the last group of letters in the signature. We trace from P_1 to P_3. P_2 and P_4 are points corresponding to P_1 and P_3 found by using the gradient direction. The widths from P_1 (or P_2) to P_3 (or P_4) are all greater than T. However, the corresponding points are not on the same loop. In this case, the points on the contour from P_1 to P_3 and from P_2 to P_4 are taken as the stroke segments, shown as the thicker curves in Figure 3.9(b).

The third type of collapsed loop is shown in Figure 3.9(c). It is an isolated "o" that has collapsed into a dot. It does not occur in the signature shown here, but it does arise in some cases. We start at an arbitrary point on the edge; for that point, say P_1, the situation is the same as in detecting the first type of collapsed loop, shown in Figure 3.9(a). Therefore, we
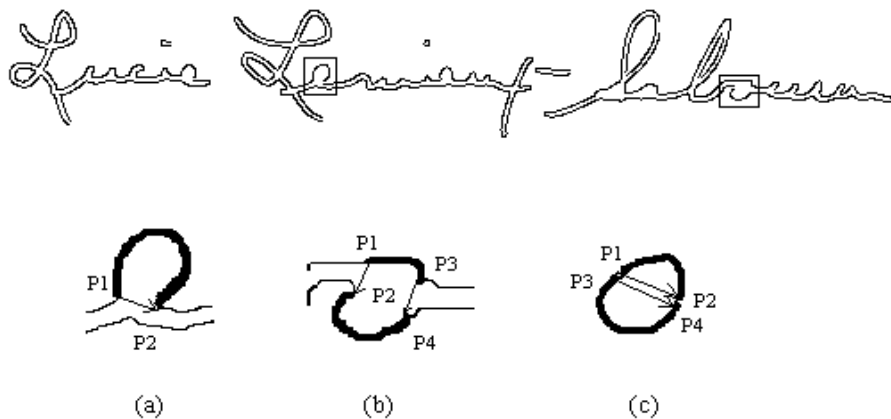
27

Figure 3.9: Illustration of getting the stroke segments of a collapsed loop.

obtain a stroke from P_1 to P_2. We continue with the neighboring point, say P_3, and get another segment (P_3 to P_4). As the thicker curve shows in Figure 3.9(c), we have found two segments. We perform a merging process, as discussed in Section 3.4.1, to merge these segments.

By proceeding in this way, we are able to restore approximately 80% of the collapsed loops. For double collapsed loops, like the last stroke of the "S" in the signature in Figure 3.6(a), the tiny loop in the original signature is merged with a larger loop in the edge image (Figure 3.6(c)), and our approach does not work. However, when we make the correspondence as discussed in Section 3.5, we will associate a reverse stroke with each stroke segment. The segment obtained from the merging of the loops is the one that goes from right to left, and the reverse segment comes back from left to right. The loop that was lost at this stage can thus be recovered, at least approximately.

Merging Fragmented Segments

Due to inaccuracy in detecting junction points from changes in gradient direction, as well as discontinuities in the detected edges caused by noise, small pieces that should belong to a single stroke segment are sometimes not connected together. Merging these pieces before making a correspondence between the model and the questioned signature can increase the speed of matching and enhance its accuracy.

One way of doing this is to merge two neighboring pieces when the stroke direction at the end of one stroke is consistent with that at the beginning of the second stroke. This is, of course, sensitive to noise. It is possible that the two directions are slightly offset due to noise, or that two pieces that satisfy the condition are not supposed to connect to each other (e.g., the bar in a "t").

To make the process less sensitive to noise, we refer back the original image. If the two neighboring pieces are separated by a junction point, the grey levels in the region between the ends of the pieces are lower due to the presence of dark pixels as in Figure 3.10(a). The grey levels in the region between two neighboring pieces that should be merged into one segment are not as dark (Figure 3.10(b)). Thus, we can check the grey levels in this region to see if the two segments should be merged.

Another merging method we use refers back to the edge image. If both edges of the neigh-
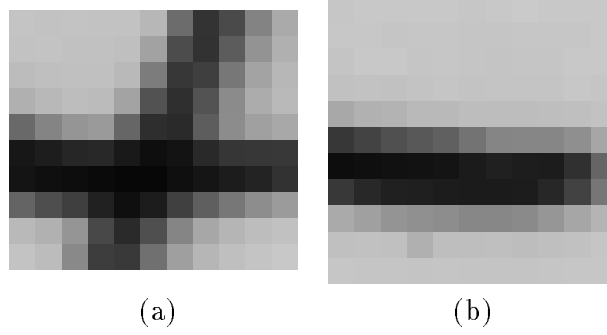
Figure 3.10: (a) The pieces are separated by a junction point (b) The pieces belong to one segment.
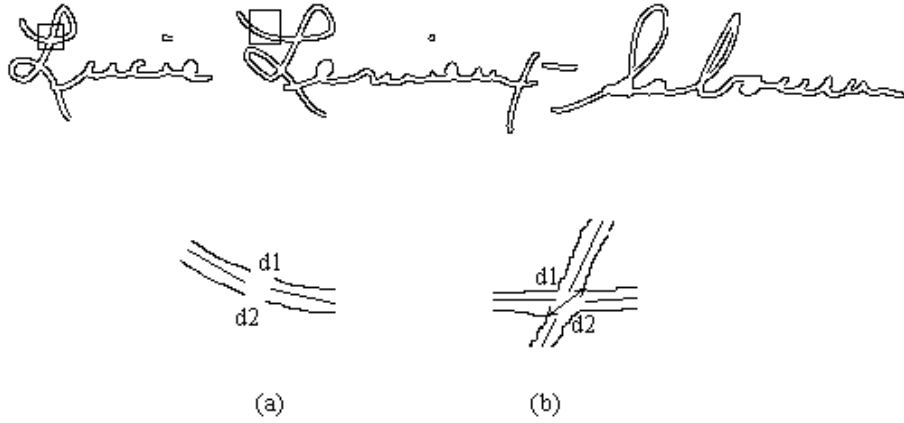
Figure 3.11: (a) Two pieces belong to one segment. (b) Two pieces are separated by a junction point.

boring segments are either connected or close enough, we merge the two segments. If the two neighboring segments are separated by a junction point, the edges will usually be not close enough, at least on one side, for them to be connected. As shown in Figure 3.11(a), when $d_1 \approx d_2$ and $d_1, d_2 \leq T$ (T is a threshold), we merge the two segments. In Figure 3.11(b) $d_1 \leq T$ while $d_2 > T$, so we don't merge the segments. The edge image is again shown in Figure 3.11. This method is more accurate in correctly identifying junction points, but it is sensitive to situations in which there are large gaps in edges due to noise; an example is the gap on the outer edge of the letter "S".

3.5 Signature Matching

Our matching process involves establishing a segment-wise correspondence between the model and the questioned signature. We are not trying to detect all types of forgeries at this point, but we want to obtain the correspondence in order to examine additional features of the signature. These features are structural and include the position of each stroke segment, its shape (reflected by its curvature), and the connections between neighboring segments. We can then use gross structural features to detect random and simple forgeries. We can also examine additional local

29

features to detect skilled forgeries; this will be the subject of a subsequent chapter.

3.5.1 Cost function

A cost function is used to evaluate the quality of a match between the model and the questioned signature. The best match is the one with the lowest cost, i.e., the sum of the components defined in the cost function is the minimum among all the possibilities.

The distance (cost) between the questioned signature and model is defined as

$$d = \sum_{i=1}^{i=M} d(i)$$

where $i = 1, \ldots, M$ is the index of the stroke segments, M is the number of stroke segments in the model, and $d(i)$ is the cost of the ith stroke segment. Using the features discussed in Section 3.3, we have

$$d(i) = \sum_{k=1}^{k=K} w_k(i) * d_k(i)$$

where $d_1(i)$ is the difference in relative position between the model and questioned signature on stroke segment i, $d_2(i)$ is their difference in incoming angle, $d_3(i)$ is their difference in outgoing angle, $d_4(i)$ is their difference in average curvature, and $d_5(i)$ is their difference in size. The w_k's are the weights that were determined in Section 3.3.3; note that the weight of each feature of each segment is determined for a particular author.

Our goal is to minimize the sum of the differences between the segments in the model and the corresponding segments in the questioned signature. In doing this, we must consider many possible correspondences between the segments, and we define a NULL segment so that some of the segments can be ignored. This allows us to handle situations in which the correct correspondence is difficult or impossible to find. For example, if the model has a very different number of segments than the data, as in the case of a random forgery which requires us to match "Matthew" to "Jeff", the "att" may match well with the "eff" while the other parts of "Matthew" are all matched to NULL segments. A second example is when the model signature includes a middle initial while the questioned signature does not; here again, the initial must be matched to null segments.

To handle a third class of examples, we associate a reversed segment with each segment obtained from the segmentation process. Since the starting point of each segment was determined by scanning the image, it is quite possible that a segment in the image goes in the reverse direction from the corresponding segment in the model. When we use reversed segments, a reversed instead of a traced segment may match a segment in the model. For a retraced stroke, both the traced segment and its reversal should match the segments in the model.

3.5.2 Matching procedure

The segmentation processes described above give us a segmented model and a set of stroke segments extracted from the questioned signature. There may be too many or too few of these segments. One of the byproducts of the matching process is that we can finalize the set of stroke segments and recover their order.

The average curvature, incoming and outgoing angles, and size of a NULL segment are defined as 0. Its relative position, however, is assumed to be close to the one it is supposed to match in the model. Each segment in the data can only be matched to one segment in
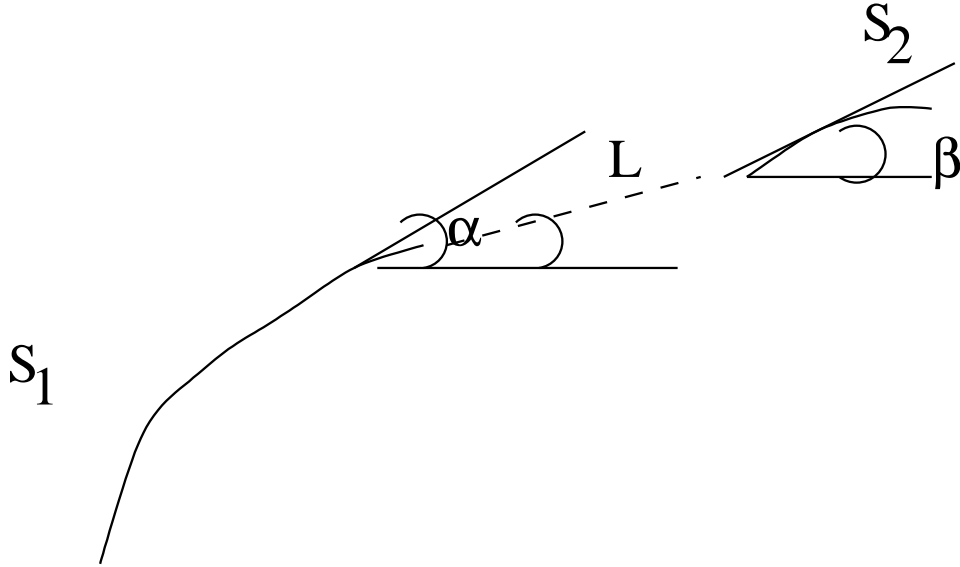
Figure 3.12: Illustration of the smoothness criterion in getting the next candidate segment.

the model. However, if a segment in the data has been matched to a segment in the model, its reversed segment can still be matched to another segment in the model; this is useful in handling retraced strokes as well as some types of collapsed loops, as discussed in Section 3.4.

The matching algorithm has features in common with dynamic programming and elastic matching. At each step, we try to match the model segment with a) the current segment alone; b) the next segment merged with the current; c) the last segment. This scheme is equivalent to plus and minus 1 in elastic matching. As in dynamic programming, only the best route so far to the specific segment will be remembered.

Since the stroke segments in the model are ordered, we obtain an ordering of the stroke segments in the data using the correspondence between the two. Knowing the correspondence, we can compute additional features.

Searching for the Next Candidate segment

From the current segment, we search for the next candidate segments. The searching for the next segment, the segment that can be merged with the current segment or be matched to the next model segment, is based on two criteria; one is the smoothness assumption about writing, and the other is the model.

<u>Smoothness</u>

Since the connection between stroke segments should be smooth in a genuine signature, we make the assumption that the outgoing angle of the current stroke segment should not be substantially different from the incoming angle of the next segment.

In Figure 3.12, the current segment and the next segment are denoted by S_1 and S_2 respectively. We call the line segment connecting the ending point of S_1 and the starting point of S_2 L. α is the out angle of S_1, while β is the in angle of S_2. θ is the in and out angle of L. (The in and out angle of L are equal since it is a straight line.) When the transition between the two stroke segments is smooth, the angles defined above should be close to each other in value.

31

4	3	2
5	O	1
6	7	8

Figure 3.13: The search space defined in the model.

The criterion for smoothness is as follows [71]:

$$\cos(\alpha - \theta) \times \cos(\beta - \theta) \leq T$$

where T is a threshold.

Model Matching

Even if smoothness is satisfied, sometimes the search for the next segment can be mistaken. Besides, we want to limit the search space for the next segment. This can be achieved by using model information. Model information is especially useful in the case when the stroke takes a turn that does not satisfy the smoothness criterion or when too many segments satisfy the smoothness criterion. We divide the search space as in Figure 3.13. The "O" represents the current position. The directional code represents the direction pointing to the next segment.

Searching for the next segment is based on the directional information provided by the model. Only the segments that lies within 90 deg of the model direction can be considered as candidate segments and penalties are imposed according to the offset of the direction.

Example of Matching

In this section, we give one partial example of the matching process. We show the steps of matching in Figure 3.14. In the figure, the best path for each step is also illustrated.

In this example, the small segments in the model were removed. Only major stroke segments are displayed. The numbers on the sides of the stroke segments matched at each step are the cost components. Among the numbers, the last figure is the total cost of that segment matched to the corresponding model segment. As depicted in the figure, there is a mismatch at one step. The first line of numbers shows the segment that should be matched while the second set of numbers shows the best match at that step. What is happening here is due to the noise on the edge making the traced line go down and then come back up again. This makes the outangle turn 180 deg. The outangle for the right segment has a bigger value, as expected. At the next step, some other segment has to match with the model since the one that matches best is taken. Additionally, with the use of the model connection information, the path that continues from the best path at the last step is not a best match (least cost) any more. The algorithm corrects the matching and comes back on the right track. Another point worth mentioning is that even

model

after segmentation

	0.01	0.05	0.11	0.37	0.21	0.36	1.09
	0.03	0.01	0.06	0.12	0.23	0.37	0.83
	0.01	0.03	0.05	0.12	0.26	0.23	0.71
	0.01	0.03	0.02	0.26	0.14	0.31	0.76
	0.01	0.02	0.02	0.07	0.12	0.13	0.37
	0.02	0.05	0.08	0.14	2.48	0.46	3.22
	0.00	0.19	0.02	1.19	0.17	0.35	1.92
	0.02	0.03	0.04	0.35	0.42	0.23	1.08
	0.00	0.05	0.07	0.30	0.14	0.41	0.97

after matching
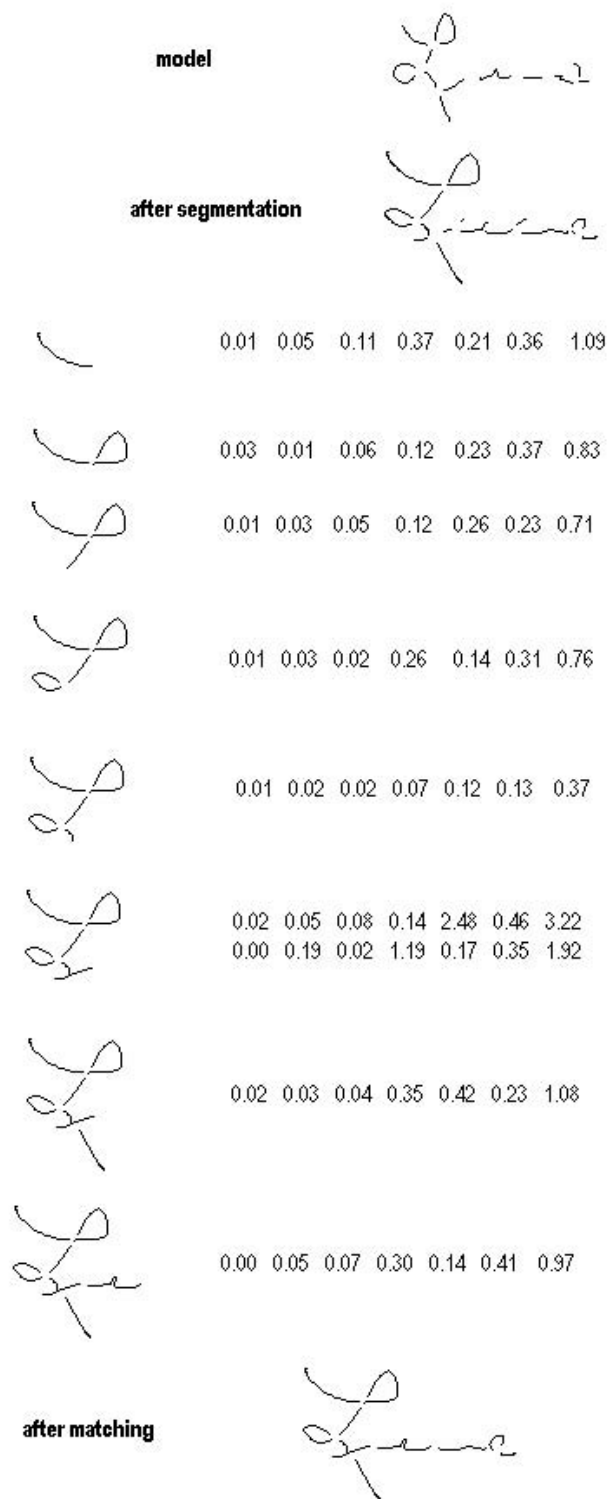
Figure 3.14: A partial example of the matching procedure.

33

though the model and the questioned signature have minor differences, two of the segments were merged together to be matched to a model segment.

3.5.3 Random forgery detection

In the process of making local correspondence, we have calculated the cost function and have estimated the final cost for the entire signature. A genuine signature should correspond well with the model, and its cost should be smaller than a threshold determined from the reference signatures even if we allow for variation in a person's writing. A random forgery, generated without knowing the spelling of the signature, will not match the structure of the genuine signature, and will have a final cost that is much higher. In the next section, we will discuss our experimental results on random forgery detection.

3.6 Experiments

In this section, we discuss experimental results using a a signature database provided by Prof. Robert Sabourin of the École de Technologie Supérieure in Montreal, Canada. The database consists of 800 hard copy signatures written by 20 signers, i.e., 40 signatures of each writer.

3.6.1 Methodology

A traced model is first generated for the writer whose signature will be verified from one of the hard copy signatures. We divide the signatures of each writer into two sets, the training set S_{training} and the testing set S_{testing}. The training set is used for generating the weights on the features for each stroke segment as discussed in Section 3.3.3.

We use a simple binary testing scheme. If the cost for corresponding a questioned signature to the model is greater than a pre-determined threshold, we conclude that the questioned signature is a forgery. Otherwise, we accept it as a true signature. A threshold is determined for each model from the training set; the process for obtaining the threshold will be discussed in Section 3.6.2.

3.6.2 Training

Model Training

One part of the model is a representation of spatial pen movements which can be gathered on-line or traced from a hard copy in advance. In our experiments, we manually traced one hard copy signature image to obtain the temporal information of the model for each writer using a mouse. The signature image for tracing was chosen to be the first signature in the signature image set for each writer. After we obtain the temporal information of the model, we perform the whole procedure of making the correspondence by choosing the weights on the features so that the cost components on each feature are approximately in the same range, e.g., the range $0 - 1$. We calculate the standard deviation of the cost of each feature for each stroke segment. We then take the reciprocal of the standard deviation as the weight. In case the standard deviation is zero, which almost surely happens only when the cost component value is zero for that stroke segment, we make the weight zero as the feature is meaningless. For example, for a small segment which is not long enough to estimate the inangle and outangle, the value of the inangle and outangle are treated as zero, which in turn causes the standard deviation to be zero.

	xposition	yposition	size	inangle	outangle	curvature
1	0.01	0.02	0.04	0.92	0.19	0.35
2	0.01	0.04	0.04	0.18	0.07	0.53
3	0.01	0.05	0.02	0.09	0.11	0.09
4	0.01	0.03	0.04	0.48	0.48	0.71
5	0.03	0.04	0.01	0.22	0.34	0.07
6	0.01	0.02	0.02	0.05	0.61	0.18
7	0.01	0.04	0.03	0.17	0.15	0.52
8	0.01	0.04	0.07	0.14	0.08	0.58
9	0.03	0.05	0.04	0.15	0.19	0.23
10	0.03	0.06	0.13	0.61	0.39	0.99
11	0.03	0.03	0.06	0.88	1.01	0.54
12	0.04	0.04	0.03	0.26	0.27	0.30
13	0.01	0.06	0.03	0.17	0.32	0.84
14	0.02	0.05	0.03	0.16	0.09	0.10
15	0.03	0.05	0.01	0.85	0.87	0.10
16	0.03	0.06	0.06	1.11	0.32	0.71
17	0.06	0.03	0.04	0.32	0.42	0.25
18	0.06	0.06	0.01	0.39	0.33	0.28
19	0.06	0.04	0.05	0.91	0.88	0.62
20	0.06	0.03	0.05	0.56	0.57	0.60
21	0.05	0.04	0.09	0.19	0.17	0.81
22	0.04	0.04	0.03	0.13	0.27	0.32
23	0.04	0.05	0.05	0.21	0.53	0.50
24	0.04	0.03	0.06	0.19	0.13	0.22
25	0.02	0.07	0.03	0.46	0.52	0.48
26	0.02	0.07	0.05	0.24	0.22	0.07
27	0.08	0.05	0.03	0.17	0.05	0.14
28	0.05	0.06	0.05	0.27	0.25	0.19
29	0.02	0.08	0.07	0.72	0.59	1.08
30	0.05	0.04	0.05	1.15	1.27	0.48

Table 3.1: Examples of standard deviations of features. The training set size in the examples is 10.

	xposition	yposition	size	inangle	outangle	curvature
1	0.01	0.03	0.04	0.96	0.17	0.36
2	0.02	0.04	0.05	0.23	0.36	0.58
3	0.01	0.05	0.02	0.36	0.35	0.10
4	0.01	0.03	0.04	0.66	0.43	0.62
5	0.03	0.05	0.02	0.41	0.54	0.07
6	0.02	0.04	0.03	0.34	0.63	0.25
7	0.02	0.03	0.03	0.16	0.18	0.44
8	0.02	0.03	0.06	0.15	0.56	0.52
9	0.03	0.04	0.04	0.14	0.16	0.20
10	0.04	0.06	0.12	0.60	0.66	1.03
11	0.05	0.03	0.05	0.93	0.94	0.50
12	0.05	0.05	0.03	0.29	0.51	0.33
13	0.02	0.06	0.03	0.28	0.30	0.72
14	0.02	0.05	0.05	0.27	0.08	0.54
15	0.03	0.06	0.01	0.72	0.71	0.25
16	0.03	0.05	0.06	0.97	0.35	0.59
17	0.05	0.03	0.04	0.31	0.39	0.21
18	0.05	0.09	0.02	0.40	0.38	0.38
19	0.05	0.05	0.05	0.82	0.79	0.52
20	0.05	0.04	0.04	0.65	0.56	0.57
21	0.04	0.04	0.08	0.22	0.26	0.73
22	0.03	0.03	0.03	0.20	0.32	0.39
23	0.04	0.04	0.06	0.22	0.45	0.50
24	0.03	0.03	0.06	0.16	0.14	0.44
25	0.02	0.07	0.04	0.44	0.49	0.60
26	0.02	0.06	0.05	0.25	0.38	0.39
27	0.06	0.05	0.03	0.32	0.13	0.18
28	0.04	0.05	0.06	0.37	0.31	0.20
29	0.02	0.07	0.11	0.59	0.61	1.40
30	0.05	0.06	0.04	0.97	1.05	0.45

Table 3.2: Examples of standard deviations of features. The training set size in the examples is 20.

Table 3.1 and Table 3.2 give examples of trained weights for writer A using training set of 10 and 20, respectively. Let us examine Table 3.1. The values of the weights are only for that specific feature. The standard deviations are relative when compared between the features. For example, the standard deviation for x-position is generally smaller (the entries in the table are the values of the standard deviation) than that for y-position. One reason for this is that the rectangular shape of the signature makes the difference value of the x-position calculated as defined in Section 3.3.2 smaller than the difference value of the y-position. On the other hand, when there is any skew of the signature when it is scanned in, the skewness effects the y-position more than the x-position. As we observe from Table 3.1, most of the stroke segments are more or less equally consistent on the features of position and size, due to the restrictions placed when generating the database, except for a few stroke segments which are less consistent in size, and even fewer in position. It can be observed in the database that those stroke segments are indeed less consistent. Since the x-position and y-position, as well as the size, are rather consistent, the standard deviations for these features are relatively small so that the inverses can become very large. For example, the standard deviation of the x-position and y-position features for the first stroke segments in Table 3.1 are both very small, meaning they both are rather consistent. They should be treated almost equally especially taking into consideration the argument that the standard deviation calculated for y-position is generally slightly larger than that for x-position. However, by using the reciprocal of the standard deviation, their difference can be enlarged 50 times. The same argument applies to the size feature. To make the weights less sensitive, we add 1 to all the nonzero standard deviations before we use them as the weights for the features. By doing this, features with deviations in the 1/100 range can be treated similarly.

Threshold Training

A threshold is calculated based on the distribution of the costs of the signatures in the training set. In Figures 3.15, we show the costs of the reference signatures of writer A and the forgeries. The thresholds are determined from the means and standard deviations of these costs.

In this section, we discuss two methods of training the thresholds. One is based on the mean and standard deviation of the training set of the questioned signature as well as those of the other writers' signatures in the database. The other one is based solely on the mean and standard deviation of the training set of that person's signature. The latter, however, needs an experiment to obtain optimized parameters.

<u>**Method I**</u>

We assume that the match cost distribution of the kth training set of genuine signatures when matched to that writer's model is Gaussian with mean m_k and standard deviation σ_k. (We also assume that these normal distributions are independent.) Let C_{kk}^i be the cost of matching the ith signature in the training set of writer k to that writer's model. These costs are distributed as $N(m_k, \sigma_k)$, where

$$m_k = \frac{1}{N_{\text{train}}} \sum_{i=1}^{N_{\text{train}}} C_{kk}^i$$

$$\sigma_k^2 = \frac{1}{N_{\text{train}} - 1} \sum_{i=1}^{N_{\text{train}}} (C_{kk}^j - m_k)^2$$

37
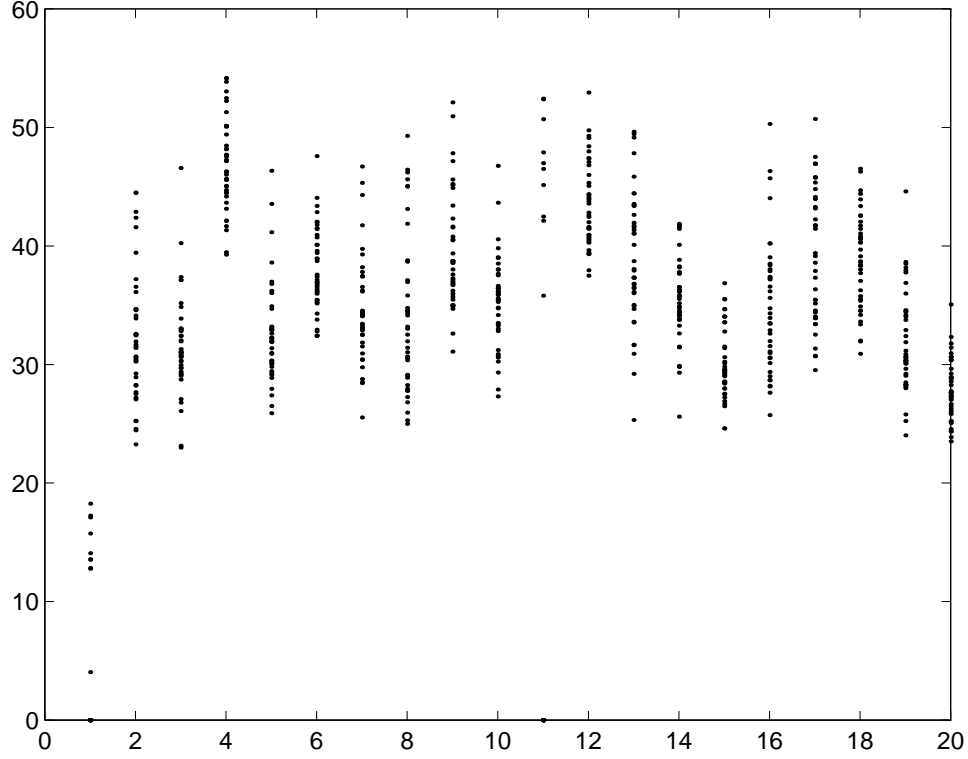
Figure 3.15: The cost distribution of the training signatures of writer A.

and the Gaussian distribution is

$$f_k(x) = \frac{1}{\sqrt{2\pi\sigma_k^2}}\exp\{-\frac{(x - m_k)^2}{2\sigma_k^2}\}$$

For the rest of the writers in the database, the cost distribution $g(x)$ of comparing their signatures to any individual writer's model will also be assumed to be Gaussian with mean M and standard deviation Σ, given by

$$M = \frac{1}{n}\sum_{i=1,i\neq k}^{N} N_{\text{train}}m_i$$

where $n = \sum_{i=1,i\neq k}^{N} N_{\text{train}}$, N is the number of writers in the database, and

$$\Sigma^2 = \frac{1}{n-1}((n - r)S_0^2 + (r - 1)S_A^2)$$

where $r = N - 1$ is the number of writers other than the given one, and

$$S_0^2 = \frac{1}{n-r}\sum_{i=1,i\neq k}^{N} (N_{\text{train}} - 1)\sigma_i^2$$

$$S_A^2 = \frac{1}{r-1}\sum_{i=1,i\neq k}^{N} N_{\text{train}}(m_i - M)^2$$

38

(Note that we are assuming that the database is large enough to be representative of the population of random forgeries.) The distribution $g(x)$ is then

$$g(x) = \frac{1}{\sqrt{2\pi\Sigma^2}}\exp\{-\frac{(x-M)^2}{2\Sigma^2}\}$$

Under our Gaussian assumptions, the total error probability (both type I and type II: false acceptance and false rejection) for threshold x is $P = P_{FA} + P_{FR}$, where

$$P_{FA} = \int_{-\infty}^{x} g(x)dx$$

$$P_{FR} = \int_{x}^{\infty} f(x)dx$$

so that

$$P = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi\Sigma^2}}\exp\{-\frac{(x-M)^2}{2\Sigma^2}\}dx + \int_{x}^{\infty} \frac{1}{\sqrt{2\pi\sigma_k^2}}\exp\{-\frac{(x-m_k)^2}{2\sigma_k^2}\}dx$$

(This is under the assumption that the prior probabilities of false acceptance and false rejection are the same.) To minimize the sum of the two errors we differentiate P and set the derivative equal to 0; this gives us

$$-\frac{(x-M)^2}{2\Sigma^2} + \frac{(x-m_k)^2}{2\sigma_k^2} = \ln\frac{\Sigma}{\sigma_k}$$

The solution x of this equation is the optimum theoretical threshold. (In Figure 3.16, this threshold is at the point where the two curves cross.) The advantage of computing these theoretical thresholds is that they give us an estimate of what the error rates might be if we went beyond the limits of our database.

The theoretical error rate can be calculated using

$$P_{FA} = \int_{-\infty}^{x} g(x)dx$$

$$P_{FR} = \int_{x}^{\infty} f(x)dx$$

where x is the calculated threshold.

We show the statistic values for the example of model A in Table 3.3.

The threshold for a training set of 20 is shown in Figure 3.17. As seen in the figure, the threshold can well separate the author's signature from the random forgeries. Thus the type I and type II errors for this case are 0% for this model. We can also calculate the theoretical error rate using the statistical model; the type I and type II errors are 2.5% and 0.99% respectively. The advantage of being able to calculate the theoretical values of the type I and type II error is that it gives us an estimate of what the error rate might be in the general case, i.e., going beyond the limits of the database.

In Figure 3.16, we show the two Gaussian distributions of the signature and forgeries. The crossing point of the two curves is the place where the threshold is chosen. The standard deviation of the signatures is smaller than the standard deviation of the forgeries as expected.

Method II

39

training set	10	20
M	35.9453	35.5485
Σ	6.9367	6.9071
m_k	13.9188	14.7283
σ_k	3.9950	3.1099
threshold	22.6503	21.9885
P_{FA} (actual)	0.013%	0%
P_{FR} (actual)	0%	0%
P_{FA} (calculated)	2.74%	2.5%
P_{FR} (calculated)	1.43%	0.99%

Table 3.3: The statistic values for model A.
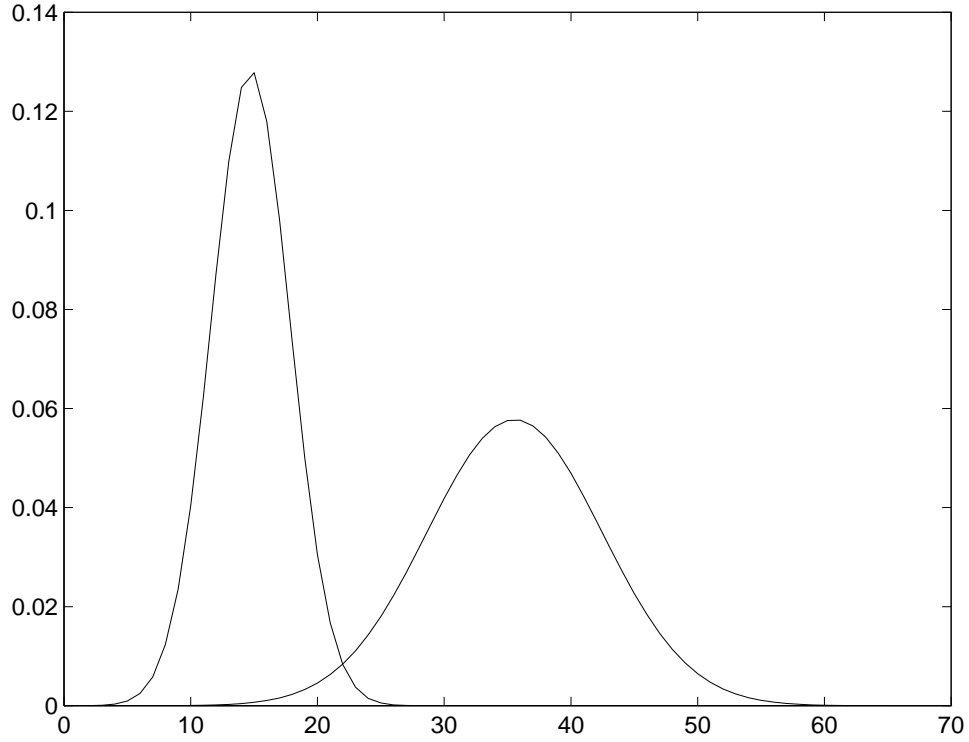
Figure 3.16: The statistical model for the signatures and forgeries of writer A.

Training set	b	Threshold	P_{FA}	P_{FR}
10	1.7	20.7103	0%	6%
10	2.0	21.9088	0%	3%
10	2.1	22.3083	0%	0%
20	1.7	20.0151	0%	6%
20	2.0	20.9481	0%	3%
20	2.1	21.2591	0%	0%

Table 3.4: Threshold calculation using training signatures only.

Training/test set size	False Rejection	False Acceptance
10/30	0.99%	0.59%
20/20	0.80%	0.45%
40/40	0.67%	0.30%

Table 3.5: Error rate vs. training set.

In this section, we consider only the training set for the model to which the questioned signature is compared. Regardless of what the distribution of the cost is, we take a linear combination of the mean and the standard deviation of the training set of the model. The calculation of the threshold is as follows:

$$x = a * m_k + b * \sigma_k$$

where a and b are constants. The calculated threshold and the corresponding results are shown in Table 3.4.

Testing Results

To give an overall idea of the cost distribution, we show the costs of all the signatures vs. model A in Figure 3.17. The horizontal axis is the model (writer) and the vertical axis is the costs. Model A is the first writer. The costs for writer A are smaller than those for the rest of the signatures. In Figure 3.18, we also show the statistics of the costs for each writer corresponding to the model. The "*" represents the mean value; the interval extends one standard deviation above and below the mean. This shows clearly the range that the costs mainly fall into for each writer. We can see that writer A can be distinguished nicely.

The general results are summarized in Table 3.5. We performed the experiments using $S_{\text{training}} = 10$, $S_{\text{training}} = 20$ and $S_{\text{training}} = 40$. The reason for using $S_{\text{training}} = 40$ is to show how good the results can be. The thresholds are chosen as described in Section 3.6.2. The theoretical results are also listed in Table 3.6.

Since the weights in the cost function are obtained statistically, the larger the training set is, the more accurate the estimate is, as can be seen in Table 3.5. More discussion follows in the next section.

3.7 Discussion, Conclusions and Extensions

We have proposed a model-based approach for the verification of static (off-line) signature images and the detection of forgeries. Our approach makes use of a model that segments the
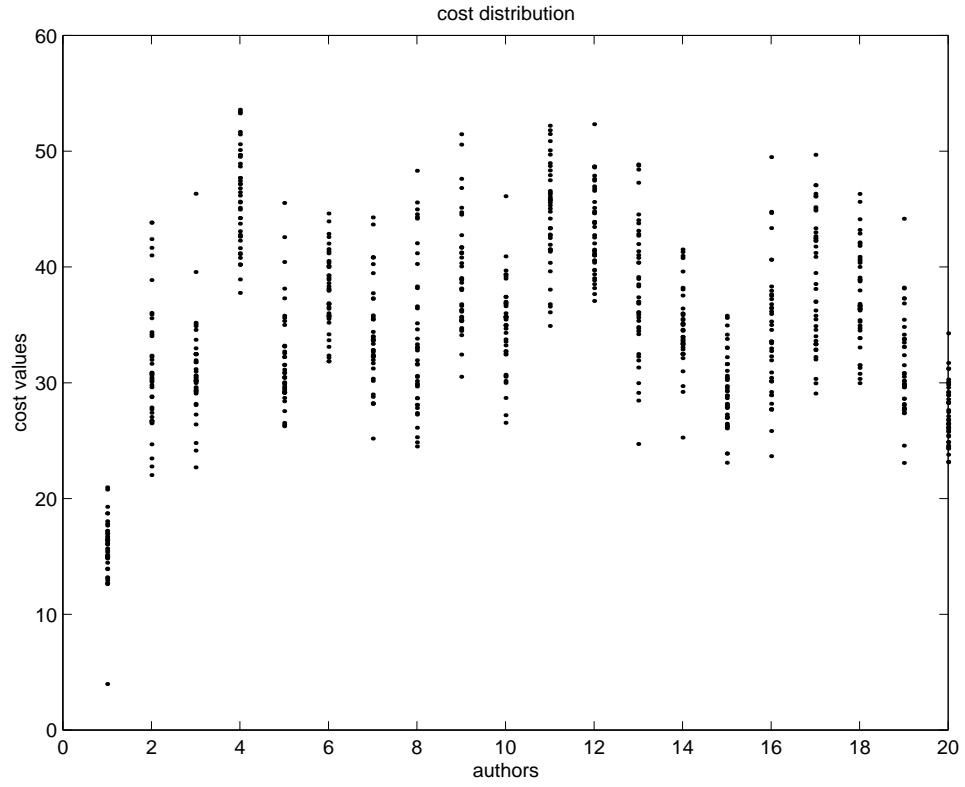
Figure 3.17: The cost distribution for all the signatures vs. model A.

Training/test set size	False Rejection	False Acceptance
10/30	2.73%	1.29%
20/20	1.66%	1.10%
40/40	1.50%	0.73%

Table 3.6: Theoretical error rate vs. training set.
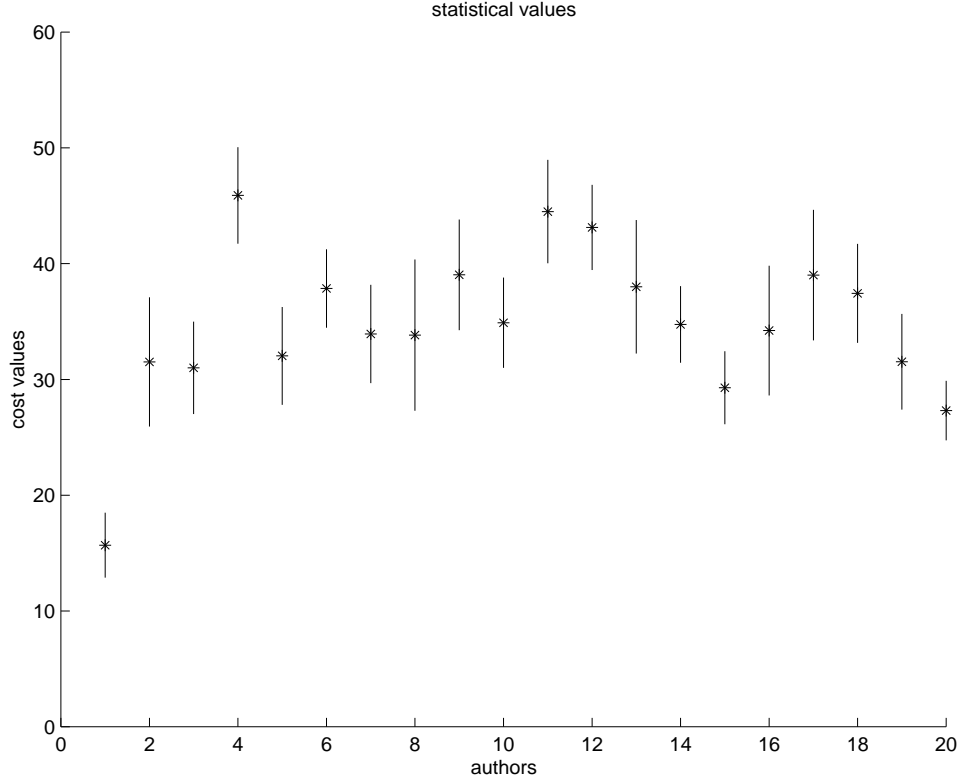
Figure 3.18: The mean and standard deviation of costs for all the signatures vs. model A.

genuine signature at junction points and attempts to follow the sequence of the writing. For verification, a questioned gray scale signature is segmented at junction points based on edge information. Features of the segments such as width, direction and type (loop, retrace, etc.) are extracted from the questioned signature image. A matching process attempts to establish a correspondence between the questioned image and the model.

We have found that our matching process works well even when the model and the questioned signature differ in size by up to 50%.

In our experiments, the segment merging process using the edge image gave better segmentation results than the process using the original image. The latter was more likely to make mistakes at junction points due to noise. These problems could be greatly reduced if we allowed the model to influence the segmentation process. The model could then guide the segmentation of the confusing parts.

If the questioned signature gives rise to a badly noise-contaminated edge image, we will not be able to get a initial segmentation, and this will affect the process of finding a correspondence between the model and the questioned signatures. Generally, the edge images obtained using the Canny edge detector are satisfactory. In some cases, gaps in the edges occur, partly because we use a fixed threshold for all the images, even if they had different contrasts due to scanning. If we use a dynamic threshold, we should be able to improve the results.

Nevertheless, we rely on a relatively clean edge image to recover the strokes. If the edge operator fails, or achieves an edge image that has a considerable number of gaps, especially on a collapsed loop, the strokes can be hard to recover. However, we have the model information which provides us with the approximate positions of critical points, loops and turns. In the next

43

chapter, we will discuss a model-guided algorithm which works directly on gray-level images. We will also modify some of the features, partly because of reliability in measurement, e.g., digital curvature, and partly because the critical points used for segmenting the strokes are different from those used in this chapter.

Chapter 4

Model-Guided Local Correspondence for Detecting Random Forgeries

4.1 Introduction

4.1.1 The problem

In this chapter, we introduce another algorithm that analyzes a questioned off-line signature by matching it to an on-line or traced model of the genuine signature, represented in the form of an ordered set of points. The matching process establishes a correspondence between the questioned signature and the model, and thus provides a basis for segmenting the signature into meaningful parts such as strokes. The total "cost" of the match (i.e., the degree of mismatch) is computed by summing the costs of matching the parts; thus both simple and random forgeries can be expected to have high costs. The segmentation is performed on gray level images as opposed to edge images as in Chapter 3. The segmentation also allows detailed information to be extracted from the questioned signature at the stroke level, and thus provides a basis for detecting skilled forgeries. The experiments reported in this chapter involve only random forgeries; applications to the detection of simple forgeries (by examining the matching costs of individual strokes) or skilled forgeries (by detailed analysis of substroke gray levels in the questioned signature) will be considered in the next chapter.

4.1.2 Our approach

This section outlines our approach.

As in our non-model-guided approach, an example of a genuine signature is used to construct a model for that person's signature. (We used only one example for each person; several models could be used for a person whose signature varies.) The model was constructed by using a mouse to trace the signature and store an ordered sequence of pixel coordinates. (Ideally, models could be constructed by digitizing on-line signatures, but on-line data was not available for the writers in our database.) The locations of endpoints, junction points, high curvature points, maxima and minima (of the y-coordinate) are also marked in the model. Further details about the models and their representation are given in Section 4.2.

For each questioned signature, we attempt to establish a pointwise correspondence between it and the model by using the model to guide a search process. Starting at the first point of the questioned signature, we search for a succession of dark points in the signature image that may correspond to the successive points of the model. To allow for differences in the size and orientation of the model and the signature, each step of the search covers an angular sector centered at the expected direction, and a range of distances up to twice the expected distance,

45

as described in Section 4.3. We also check the expected distances and directions corresponding to the previous and next model points, in case the current point of the signature is out of step with the current model point; thus the search process generates a tree of degree 3. [The search process is somewhat different at endpoints, high-curvature points, and junction points in the model, as described in Section 4.3.] When a dark point is found, the distance between its actual and expected positions is taken to be the cost of that step of the search; when no dark point is found, that branch of the search tree is terminated. This breadth-first search of a degree-3 tree is performed only to a limited depth (9, in our experiments); when depth 9 is reached, the lowest-cost path in the tree is chosen, and the search is restarted at the terminal point of that path. When the search is completed, these lowest-cost paths define a correspondence between the model and the signature.

The correspondence with the model imposes an ordering on the signature, and the endpoints and y-maxima and minima define a segmentation of the signature into substrokes. We now search for a best correspondence between the model substrokes and the signature substrokes, where the cost of a match is determined by comparing a set of geometric properties of the corresponding substrokes. The geometric properties used are x-position, y-position, x-span, y-span, size, slope histogram, and slope entropy; they are defined in Section 4.4.1. The cost is a weighted sum of the property value differences; computation of the weights is described in the next paragraph. Using this substroke match cost measure, we elastically match the sequence of model substrokes with the sequence of signature substrokes—i.e., each signature substroke is compared with the preceding, corresponding, and following model substrokes, and we find the elastic match between the sequences that has the lowest total cost, as described in Section 4.4.2.

In computing the cost of the elastic match, initially each property of each substroke is given equal weight. Using this initial cost function, the model is matched to a training set of signatures of the same person. For each geometric property of each substroke, we measure the difference between its value in the model and its value in each training signature, and we compute the standard deviation of these differences over the training set. The inverse of this standard deviation is a measure of how consistently that person makes that substroke, as regards that property. Evidently, the more consistent a property is, the more weight should be given to that property in computing the costs of the substroke matches.

Signatures of other writers can also be put into pointwise correspondence with a given model signature; of course, the costs of these correspondences will usually be high, but we can still use them to segment the signatures into substrokes. We can then find lowest-cost elastic matches between these substroke sequences and the model; again, these costs are likely to be high, while the best elastic matches between genuine signatures (the modeled writer's signatures) and the model are likely to be considerably lower. Thus it should be possible to discriminate between genuine signatures and signatures of other writers by applying a threshold to the lowest-cost elastic match. The threshold that should be used for this purpose can be estimated from sufficiently large populations of genuine signatures and signatures of other writers, if we assume that the lowest match costs for each of these populations are normally distributed. [In our experiments, the population of genuine signatures used to estimate the thresholds was the same as the training set of genuine signatures used to estimate the weights given to the substroke properties in computing the match cost.] The computation of the property weights for each writer, and of the threshold that minimizes the total error probability for each writer, is described in Section 4.5.

In Section 4.5.3 we describe experiments in which our approach was trained and tested using a database of 800 signatures (40 from each of 20 writers). Section 4.6 summarizes the contributions of this chapter and suggests directions for future research.
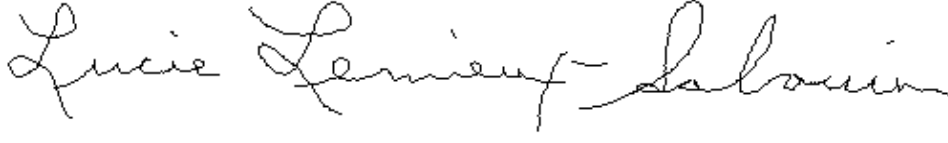
Figure 4.1: An example tracing of a signature.

4.2 Models

As discussed in Section 3.3, our model contains both spatial and temporal sequence information about the writing process. Our traced model is composed of a set of ordered points (x_i, y_i) representing samples of the pen movements. We use the sequence of distances and directions (a_i, b_i) between successive samples to represent the model. These quantities are computed as follows:

$$dx_i = x_{i+1} - x_i$$

$$dy_i = y_{i+1} - y_i$$

$$a_i = \sqrt{dx_i^2 + dy_i^2}$$

$$b_i = \tan^{-1}(dy_i/dx_i)$$

The (a_i, b_i) parameters reflect the speed and direction of the movement of the mouse when the signature was traced. The mouse position is sampled frequently, so that the coordinate differences dx_i, dy_i are at most a few pixels.

A final process involved in constructing a model is that end points, junction points, and high curvature points are labeled. The uses of these points will be discussed in later sections.

Figure 4.1 shows an example of a tracing of a signature, without explicit ordering information.

4.3 Matching a Model to a Questioned Signature

The next stage in our approach is to establish a correspondence between the model and a set of signatures, by attempting to trace each signature automatically with the guidance of the model.

Let $P_i = (x_i, y_i)$ be the ith point calculated from the model and $P_i' = (x_i', y_i')$ the corresponding point found in the questioned signature. We want to minimize the dissimilarity measure

$$F = \sum \parallel P_i - P_i' \parallel$$

$$F = \sum \sqrt{(x_i - x_i')^2 + (y_i - y_i')^2}$$

under the smoothness constraint

$$S = \sum \left| \frac{\triangle y_i'}{\triangle x_i'} - \frac{\triangle y_{i-1}'}{\triangle x_{i-1}'} \right| \leq T_1$$

47

and the darkness constraint (to ensure the point P_i' is on the signature)

$$D_i = g(x_i', y_i') \le T_2 \qquad \text{for all } i$$

To simplify the problem, we can use equal constraints; the Lagrange function can thus be written as

$$L = F + \alpha_i S + \beta_i D_i$$

i.e.,

$$L = \sum \sqrt{(x_i - x_i')^2 + (y_i - y_i')^2} + \alpha_i \{\sum |\frac{\triangle y_i'}{\triangle x_i'} - \frac{\triangle y_{i-1}'}{\triangle x_{i-1}'}| - T_1\} + \beta_i \{g(x_i', y_i') - T_2\}$$

To find the correspondence between the model and the signature, we need to solve

$$\triangle L = \triangle F + \alpha_i \triangle S + \beta_i \triangle D_i$$

This can only be solved numerically. Since the signature is generated point by point sequentially, we designed a sequential algorithm to simplify the numerical solution; this algorithm will now be described.

The model parameters (a_i, b_i) determine the positions of the successive model points. Starting from the first point (x_0, y_0) of the signature, we examine these points in the signature image and search for dark points (= points whose gray levels are below a given threshold—in our experiments, 184, determined empirically) in the vicinities of these points.

Let (x_i, y_i) be the current point that we have found in the questioned signature, and let (a_i, b_i) be the current pair of model parameters. If the questioned signature were an exact match to the model, its next point should be at $x_{i+1} = x_i + dx_i$, $y_i + dy_i$, where $t_i = \tan b_i$ and

$$dx_i = a_i \sqrt{1/(1 + t_i^2)}$$

$$dy_i = a_i t_i \sqrt{1/(1 + t_i^2)}$$

To allow for variations in the orientation of the signature, we search at distance a_i in a 90° angular sector centered in direction b_i. If the sector contains a dark point, we take its darkest point to be the next signature point, see Figure 4.2. If not, we continue to search in sectors of increasing radii; we start the search in direction b_i and search on both sides of it; if no dark point is found by the time we reach distance $2a_i$, we terminate the search (i.e., that branch of the search; see the next paragraph).

To allow for the possibility that the signature has gotten out of step with the model, this search process is conducted using not only the parameters (a_i, b_i) of the current model point, but also the parameters (a_{i-1}, b_{i-1}) and (a_{i+1}, b_{i+1}) of the preceding and following model points (except at the first step of the search, where there is no preceding point). Thus the search process is described by a tree of degree 3. Figure 4.3 is a simple example illustrating the shape variations that can be handled by allowing the search to use $(a_{i\pm1}, b_{i\pm1})$ as well as (a_i, b_i). In this example, the a's are all equal and there are only two b's (0° and 45°). The original five-step model is shown at the top of the figure; in examples (a) and (c), one step is skipped; in example (b), one step is skipped and another is repeated; in example (d), one step is repeated. (These examples show only the ideal search positions; they do not show the direction and distance tolerances described above.)

48

Figure 4.2: Illustration of searching for the next point.

Figure 4.3: Examples of shape variations.
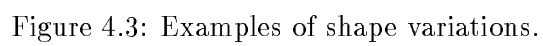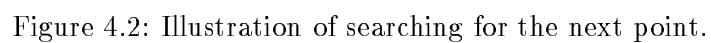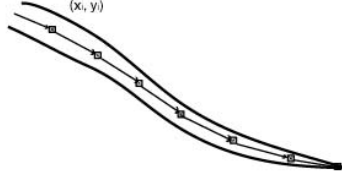
Figure 4.4: Searching at an end point.

If a dark point is found in the signature at (x_i', y_i') when we are searching for a model point at (x_i, y_i), we assign a cost to that point equal to the Euclidean distance between (x_i, y_i) and (x_i', y_i'). The total cost of a path through the search tree is the sum of the costs of the points along that path.

Since the search tree has degree 3, the total cost of searching it to depth M is $O(3^M)$; for a model containing dozens of points, this would be prohibitive. We therefore perform the full degree-3 search only to a bounded depth (in our experiments: 9). We find the lowest-cost path that reaches this depth, and restart the search at the terminal point of that path.

The search process described so far is used to search for ordinary portions of the model, but it is modified when we search for special points such as end points, junction points, and high curvature points:

a) When the search reaches an end point in the model, we first continue to search in the previous direction, in case the stroke of the signature terminating at that end point is longer than the corresponding stroke in the model. When no dark point is found, we restart the search at the position of the next model point.

   In Figure 4.4, (x_i, y_i) corresponds to an end point of the model. The search continues until the stroke terminates.

b) At a high-curvature point, the writing changes direction significantly. In order to reliably trace the signature when we reach a high-curvature point in the model, we first continue to search in the previous direction; when no dark point is found, we search in the next direction specified by the model.

   In Figure 4.5, (x_i, y_i) corresponds to a point just before a high curvature point. We continue to search in steps of length a_i in direction b_i until we reach the high curvature point in the signature. The search then continues using distance a_{i+1} and direction b_{i+1} (or, if necessary, a_{i+2} and b_{i+2}), noting that the search branch using (a_i, b_i) has been terminated.

c) Junction points may need special treatment because the tolerance in orientation may lead to ambiguity in deciding which branch to follow at a junction. However, from the model, we know where junction points are expected, and we can pick the branch that yields the smoothest trace.

   In Figure 4.6, when the search reaches point (x_i, y_i) which corresponds to a junction point in the model, multiple dark points will be encountered. For this situation, we find two dark points. We compare the direction from which we reached the current point and the directions from the current point to these candidate dark points. The candidate that corresponds to the smaller change in direction will be chosen as the next point.

50
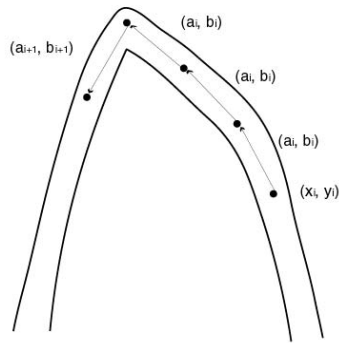
Figure 4.5: Searching at a high curvature point.
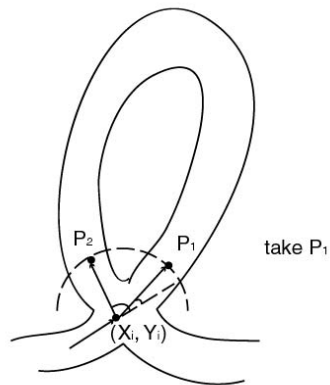
Figure 4.6: Searching at a junction point.

(a)

(b)

(c)

Figure 4.7: (a) Model "a"; (b) letters "a" with the recovered pen trace overlaid, without special treatment of junction points; (c) letters "a" when the junction points are specially treated.

In Figures 4.7-4.9, we show some results of tracing single letters to show how the algorithm handles different variations and recovers loops and retraced strokes. In Figure 4.7(b), we see that the loops of the third and fourth "a"s are not correctly recovered because the slope of the stroke at the junction point differs from the model. By special handling of junction points, we can solve this problem; the result is shown in Figure 4.7(c). In Figures 4.7–4.9 the recovered trace is overlaid on the image. Figure 4.10 shows an example involving tracing of an entire signature. This signature was not the one from which the model was derived; the differences can be seen on close inspection of the figure.

Since it is based on the (a_i, b_i)'s, our algorithm is not scale-invariant. However, because of the elastic nature of the segmentation process, the signatures need not all be exactly the same size; their sizes need only be roughly in the same range. [We could modify the parameters a_i to make them relative to the size of the signature; this would make the algorithm truly scale invariant.] In addition, because we are searching in a $\pm 45°$ angular sector, the algorithm has a considerable degree of skew insensitivity.

4.4 Segmentwise Comparison of the Model and the Signature

The pointwise correspondence between a model and a questioned signature, established as described in Section 4.3, next allows us to perform a segmentwise comparison of the model and the signature by comparing geometric properties of their corresponding segments. To perform this comparison, the model is segmented into "substrokes" at endpoints, y-maxima and y-minima. The signature is segmented into corresponding "substrokes" from the pointwise correspondence. For each segment, we measure a set of geometric properties (x-position, y-position, x-span, y-span, size, slope histogram, and slope entropy), as described in Section 4.4.1. We then compute a best match cost by elastically matching the sequences of segments, where the

(a)

(b)

Figure 4.8: (a) Model "c"; (b) letters "c" with the recovered pen trace overlaid.

(a)

(b)

Figure 4.9: (a) Model "l"; (b) letters "l" with the recovered pen trace overlaid.

(a)

(b)
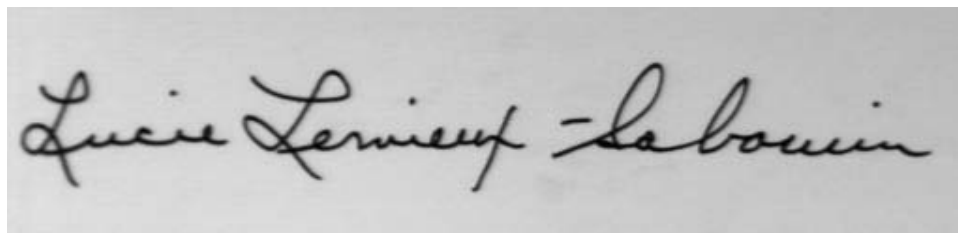
(c)

Figure 4.10: (a) A model signature; (b) a signature image; (c) the signature with the recovered pen trace overlaid.
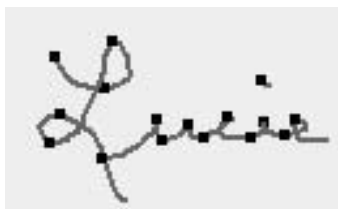
(a)   (b)   (c)

Figure 4.11: (a) A model; (b) a mismatch image; (c) the mismatch with the pen trace overlaid.

(a)   (b)   (c)
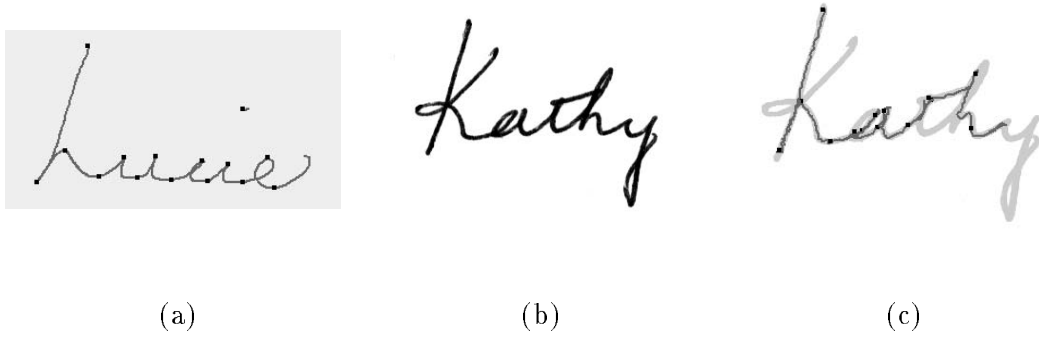
Figure 4.12: (a) A model; (b) a mismatch image; (c) the mismatch with the pen trace overlaid.

cost of a match is obtained by weighted summation of the differences of corresponding property values; see Section 4.4.2. Initially, the weights are all set to 1; but after we have matched a set of genuine signatures to the model of that writer's signature, we can estimate how variable the properties of the segments of that writer's signature are, and we can use weights that are inversely proportional to the property variabilities, as described in Section 4.4.3.

4.4.1   The properties

The segment properties we use are x-position, y-position, x-span, y-span, size, slope histogram, and slope entropy. (These last two properties were used in place of the inangle, outangle, and curvature properties of Chapter 3 because they are more stable.) These geometric properties of the segments are appropriate for detection of random or simple forgeries, where the strokes composing the signature are likely to have different positions, sizes and shapes than those in a genuine signature.

x- and y-position

The position difference properties are defined by

$$d_{\mathrm{pos}} = |S_{\mathrm{pos}} - M_{\mathrm{pos}}|$$

where S_{pos} and M_{pos} are the x- or y-positions of the segments in the questioned signature and the model respectively as defined in Section 3.3.2.

x- and y-span

The span difference properties are defined by

$$d_{\mathrm{xspan}} = |S_{\mathrm{xspan}} - M_{\mathrm{xspan}}|$$

$$d_{\mathrm{yspan}} = |S_{\mathrm{yspan}} - M_{\mathrm{yspan}}|$$

where S_{xspan} and M_{xspan} are the x-span of the segment in the questioned signature and model respectively, while S_{yspan} and M_{yspan} are the corresponding y-spans.

The x-span of a model or signature segment is the width of the bounding box of the segment normalized by the width of the bounding box of the model or signature. The y-span is defined

$$S_{\text{xspan}} = \frac{x}{X} \qquad\qquad S_{\text{yspan}} = \frac{y}{Y}$$
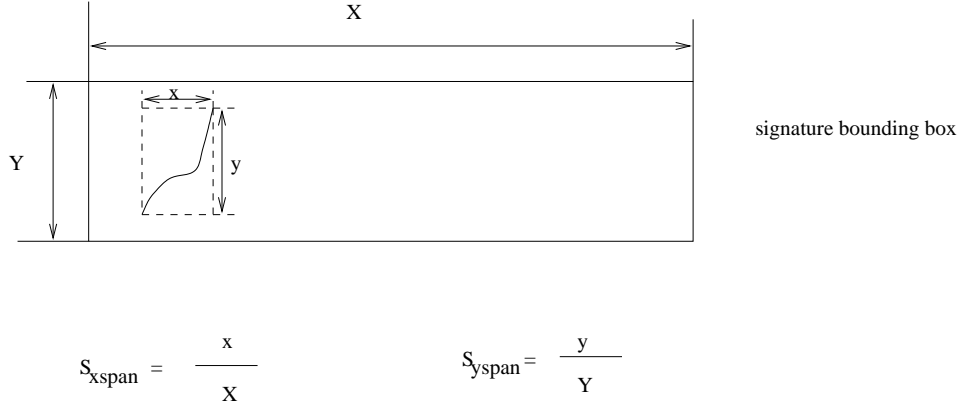
Figure 4.13: Illustration of x and y span.

as the height of the bounding box of the segment normalized by the height of the bounding box of the model or signature. See Figure 4.13 for an illustration. These properties provide information about the x- and y-axis projections of the segment. If the segment is primarily vertical, it will have a large y-span and a small x-span, while a horizontal segment will have a small y-span and a large x-span.

When segmenting the strokes at local y-minima and y-maxima, the initial/final angles used in Chapter 3 may not be as significant . x-span and y-span can also be measured more accurately than the angles and are less prone to noise.

Size

The size difference property is defined by

$$d_{\text{size}} = |S_{\text{size}} - M_{\text{size}}|$$

where S_{size} and M_{size} are the arc lengths (computed by joining successive sample points with straight line segments) of the segments of the questioned signature and model respectively as defined in Section 3.3.2.

Slope histogram

The slope histogram difference property is defined by

$$d_{\text{slope}} = |S_{\text{slope}} - M_{\text{slope}}|$$

where S_{slope} and M_{slope} are slope vectors derived from the segments of the questioned signature and model respectively.

A slope vector is an 8-component vector, where the components are the numbers of steps in the segment, from sample point to sample point, in each of the 8 directions east, west, north, south, northeast, northwest, southeast and southwest. These numbers are normalized by the total number of sample points in the segment, so that each component of the vector represents the fraction of moves that the segment makes in a specific direction. The difference between the two vectors is defined as the sum of the squares of the differences between corresponding components.

Figure 4.14 shows three examples of slope histograms. In the figure, the slope entropy values of the segments (to be discussed next) are also given. The horizontal axis gives the direction codes, defined as follows:

$$
\begin{array}{ccc}
4 & 3 & 2 \\
5 & & 1 \\
6 & 7 & 8
\end{array}
$$

The vertical axis is the value of the vector component corresponding to each direction. The first segment begins at the upper right and ends at the bottom. Thus, its histogram has a large peak in direction 7 due to its general downward direction. It also has a smaller peak in direction 6. The second segment goes from the lower left to the upper right. The histogram emphasizes directions 1 and 2, but it has a more uniform distribution because of the curvedness of the segment. The third segment also goes from the lower left to the upper right, but since it is straight, its histogram is almost entirely in directions 2 and 3.

Slope entropy

The slope entropy difference is defined as

$$
d_{\text{slopeEntropy}} = |S_{\text{slopeEntropy}} - M_{\text{slopeEntropy}}|
$$

where $S_{\text{slopeEntropy}}$ and $M_{\text{slopeEntropy}}$ are the slope entropy of the signature segment and model segment, respectively. The slope entropy is defined as

$$
E_s = \sum_{i=1}^{8} Slp[i] \log Slp[i]
$$

where $Slp[i]$ is the ith component of the slope vector defined in Section 4.4.1. The slope entropy is a measure of variability in direction. If the slopes are all in one direction, i.e., a perfect straight line, E_s will be zero, and it will be a maximum when all the components of the slope vector are equal. Similar information could be provided by measuring the curvature of the segment, but the entropy is less sensitive than curvature to the effects of digitization. Figure 4.14 gives the slope entropy values for the three segments. As expected, the second segment, which is the most curved, has the larger slope entropy.

The slope histogram and slope entropy are used as a replacement for digital curvature in Chapter 3. They have the property of measuring shape and turn, and are more robust to noise.

4.4.2 Finding the lowest cost match

The cost of a segmentwise correspondence between a model and a questioned signature can be computed by summing the differences between corresponding property values of the corresponding segments. The cost can thus be written as follows:

$$
d = \sum_{i=1}^{i=M} d(i)
$$

where M is the number of segments in the model and $d(i)$ is the cost of the ith segment. In terms of the individual property differences, we have

$$
d(i) = \sum_{k=1}^{k=K} w_k(i) * d_k(i)
$$

57
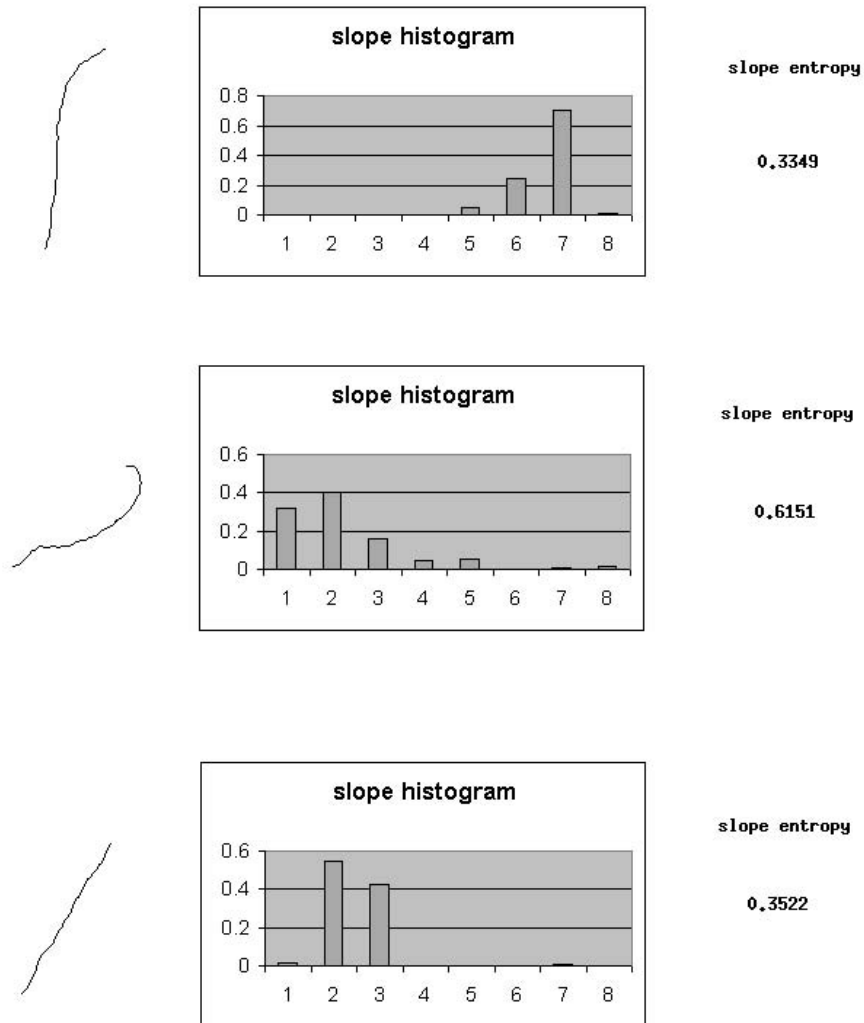
Figure 4.14: Examples of slope histogram and slope entropy.

where $d_k(i)$ is the difference between the values of the kth property for the model and questioned signature on segment i. The w_k's are weights; initially, they are all set to 1, but later they are adjusted to depend on the variabilities of the property values for the given writer, as described in Section 4.4.3.

Our goal is to find a correspondence that minimizes the sum of the differences between the segments in the model and the corresponding segments in the questioned signature. To do this, we use elastic matching, which allows many possible correspondences between the segments. Specifically, each segment of the model is compared not only to the corresponding segment of the signature, but also to the preceding and following segments. We also allow a model segment to correspond to a NULL segment of the questioned signature, or vice versa. The x-span, y-span, size, and slope entropy of a NULL segment are defined to be zero. Its relative position and slope vector, however, are assumed to be close to those of the segment that it is being matched with, so that their differences are zero. Thus the cost of matching a real segment to a null segment depends primarily on the size of the real segment.

4.4.3 Estimating the property weights

In comparing a questioned signature to a model, we would like to give more weight to properties of that writer's signature that are relatively invariant. Evidently, some properties may be more variable than others. A particular writer may not be consistent about the size of a certain signature segment, but may be very consistent about the curvature of that segment. In this case, we need to give more weight to the slope entropy of that segment than to its size. In general, the size or final slope of the last segment in a signature may be quite variable since the boundary of the page may have caused that segment to be cut short.

We have observed that the x-position of a given segment is generally less variable than the other properties of the segment. Over many years of signing one's name, the relative x-positions of the segments become relatively stable. When one's signature is larger, the size of each segment is likely to increase accordingly. The relative y-positions, on the other hand, are not as invariant as the x-positions. For example, the heights of tall letters such as "t" and "f" may vary. It should also be pointed out that when we normalize the position features, we divide them by the width and height of the signature respectively. The width is considerably larger than the height in most western names; hence a given amount of change in the x-position will be demphasized relative to the same amount of change in the y-position.

As another example, not all features are equally meaningful for small segments; e.g., the slope entropy may be close to 0 for a small segment unless it is very sharply curved. However, even if small segments have small property values, they do not necessarily contribute less to the final cost than do large segments. As an example, in Figure 4.15, the segments of a signature are listed in ascending order of the total costs of matching them to the corresponding model segments. The pairs of corresponding segments are shown in the first two columns. The third through ninth columns are the matching costs for x-position, y-position, x-span, y-span, size, slope histogram and slope entropy, respectively, and the last column is the total cost. As we can see from the figure, not all the small segments occur near the beginning of the list. The differences between the shapes of corresponding pairs of segments provide good illustrations of the flexibility of our matching process.

Figure 4.16 shows five samples of three stroke segments (nos. 1,4, and 8) of one writer's signature. The first sample is the model, which is also repeated in the first column of all five samples. The cost values of matching each segment with the corresponding model segment are listed in the same order as in Figure 4.15. Table 4.1 shows the standard deviations of the costs

7	/	)	0.02	0.03	0.01	0.03	0.02	0.02	0.04	0.17
21			0.08	0.04	0.01	0.05	0.02	0.01	0.03	0.23
0			0.00	0.04	0.00	0.01	0.00	0.08	0.10	0.25
20			0.08	0.07	0.01	0.05	0.03	0.02	0.01	0.27
6			0.01	0.00	0.02	0.02	0.02	0.09	0.16	0.32
2			0.01	0.01	0.02	0.07	0.06	0.12	0.04	0.34
5			0.02	0.05	0.01	0.00	0.00	0.01	0.25	0.35
1			0.00	0.01	0.00	0.04	0.06	0.01	0.24	0.37
10			0.00	0.13	0.02	0.03	0.02	0.03	0.18	0.41
15			0.03	0.01	0.02	0.01	0.03	0.29	0.04	0.42
19			0.11	0.02	0.04	0.10	0.08	0.01	0.08	0.43
14			0.01	0.13	0.00	0.20	0.07	0.05	0.04	0.50
23			0.03	0.02	0.04	0.28	0.10	0.01	0.03	0.51

Figure 4.15: Illustration of the cost distribution among large and small segments.

13		0.01	0.10	0.03	0.17	0.12	0.03	0.07	0.53
4		0.02	0.07	0.02	0.05	0.04	0.09	0.27	0.55
8		0.05	0.11	0.03	0.25	0.10	0.01	0.02	0.57
17		0.04	0.03	0.00	0.03	0.03	0.05	0.44	0.62
18		0.08	0.07	0.09	0.08	0.10	0.05	0.24	0.70
3		0.02	0.06	0.01	0.02	0.02	0.15	0.44	0.71
12		0.04	0.11	0.02	0.10	0.07	0.15	0.24	0.74
9		0.04	0.08	0.02	0.38	0.20	0.02	0.04	0.79
22		0.06	0.03	0.02	0.18	0.05	0.25	0.23	0.81
16		0.03	0.01	0.01	0.03	0.00	0.32	0.41	0.81
11		0.01	0.12	0.03	0.03	0.05	0.62	0.56	1.43
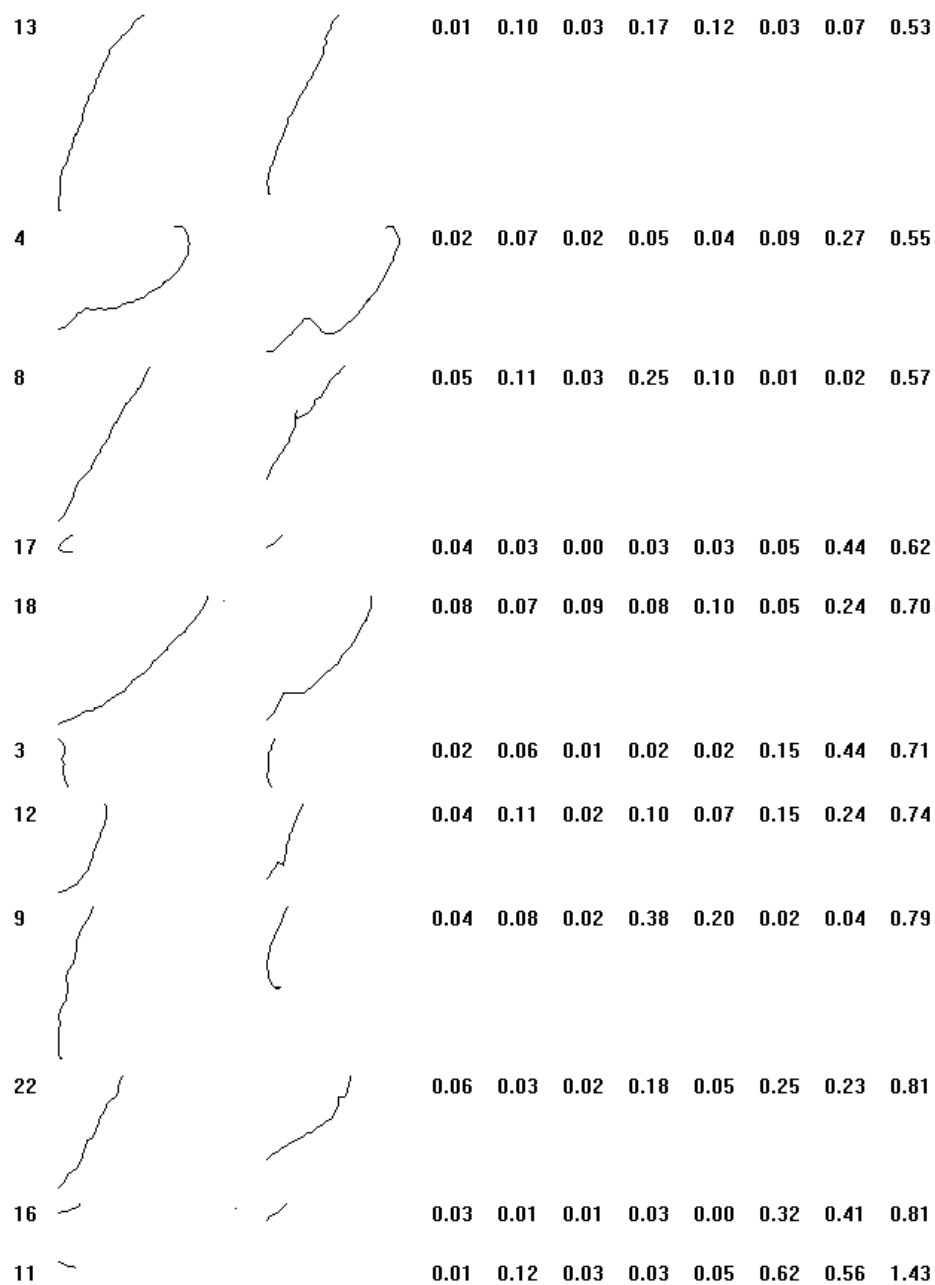
Figure 4.15, continued: Illustration of the cost distribution among large and small segments.

1		0.00	0.06	0.01	0.04	0.03	0.00	0.08
4		0.01	0.00	0.00	0.01	0.03	0.00	0.08
8		0.02	0.06	0.00	0.02	0.01	0.03	0.19

(a)

1		0.01	0.04	0.02	0.05	0.02	0.03	0.23
4		0.03	0.05	0.04	0.07	0.04	0.07	0.50
8		0.02	0.01	0.01	0.18	0.01	0.01	0.23
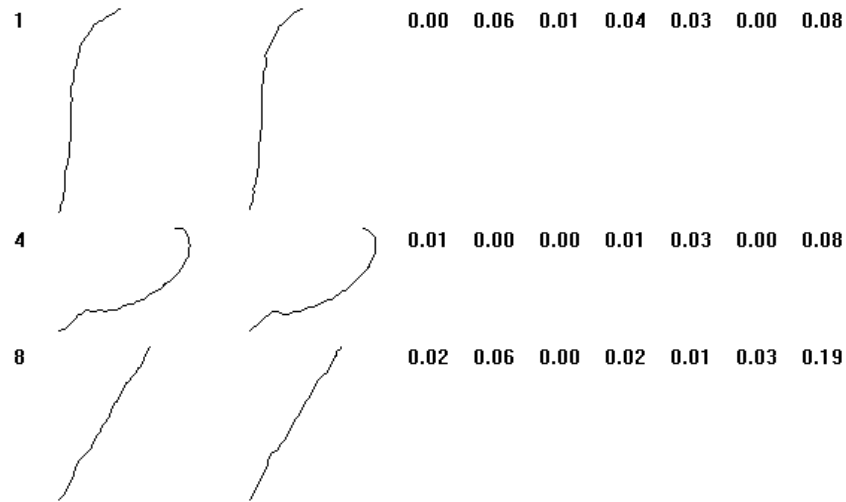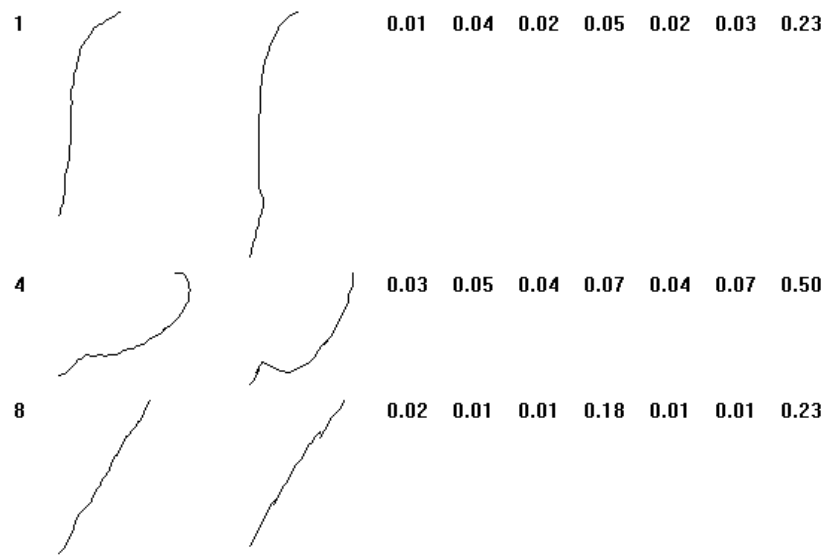
(b)

Figure 4.16: Illustration of the variability of property values.
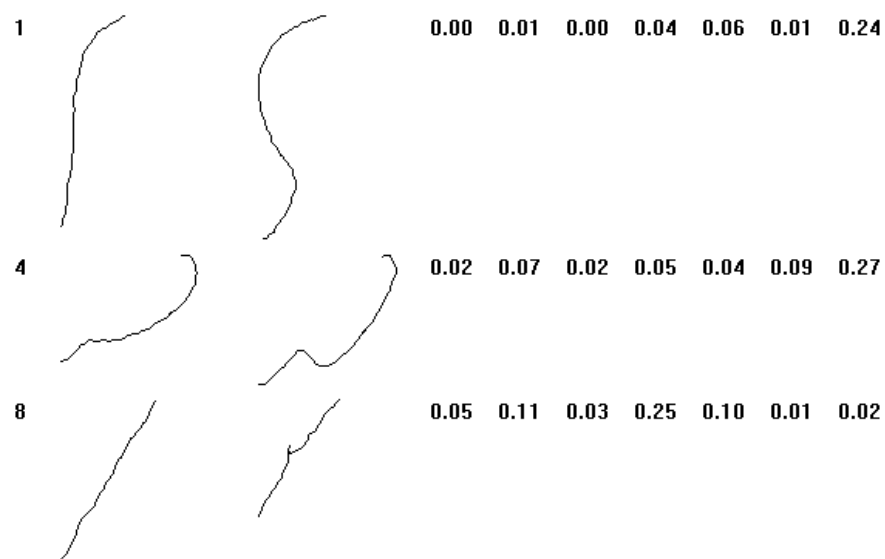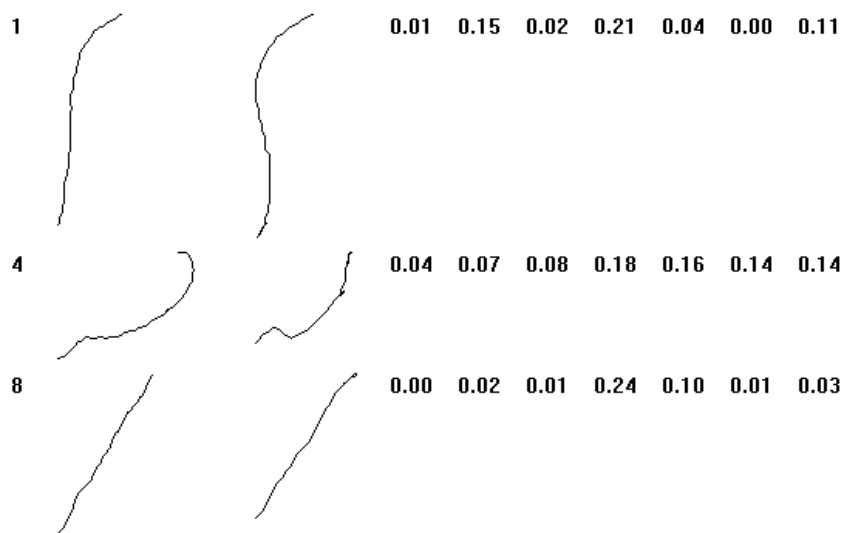
1	0.00	0.01	0.00	0.04	0.06	0.01	0.24
4	0.02	0.07	0.02	0.05	0.04	0.09	0.27
8	0.05	0.11	0.03	0.25	0.10	0.01	0.02

(c)

1	0.01	0.15	0.02	0.21	0.04	0.00	0.11
4	0.04	0.07	0.08	0.18	0.16	0.14	0.14
8	0.00	0.02	0.01	0.24	0.10	0.01	0.03

(d)

Figure 4.16, continued: Illustration of the variability of property values

1	0.00	0.02	0.03	0.00	0.02	0.05	0.28
4	0.06	0.13	0.05	0.03	0.02	0.11	0.52
8	0.03	0.13	0.01	0.08	0.05	0.02	0.06

(e)
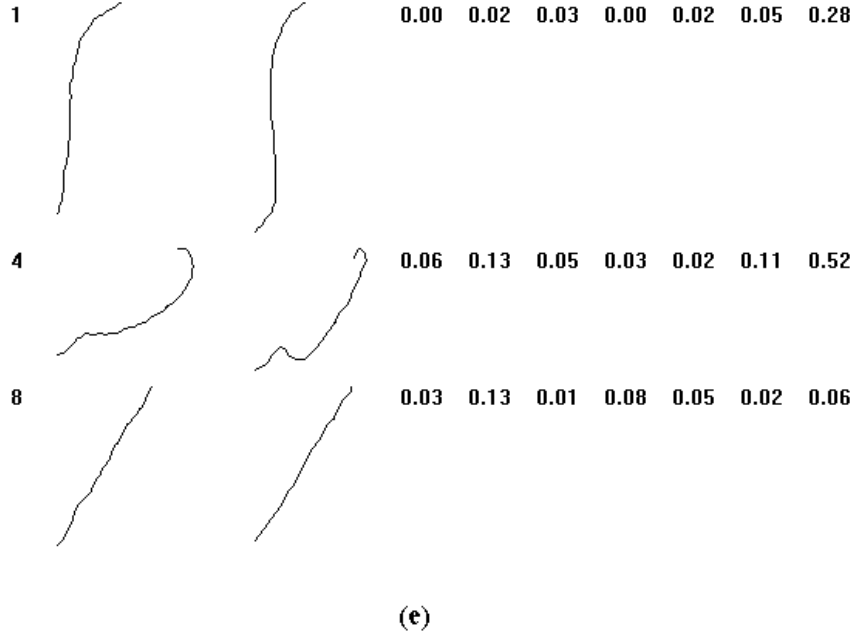
Figure 4.16, continued: Illustration of the variability of property values

for each property of each segment. (In comparing the quantities in the table to the apparent variabilities of the segments, it should be remembered that the size and span properties are normalized by the diagonal, width, and height of the entire signature.)

To estimate the variability of the properties of the segments of a writer's signature, we use a training set S_{train} of N genuine signatures. For each signature S_j in S_{train}, we compute the differences between the property values of the segments of that signature and the corresponding segments of that writer's model. We refer to these differences as "costs".

As in Chapter 3.3.3, let $\text{Cost}_j^k(i)$ be the cost of property k of the i th segment of signature S_j.

$$\text{Std}_k(i) = \frac{1}{N}\sqrt{\sum_{j=1}^{N}(\text{Cost}_j^k(i) - \frac{1}{N}\sum_{j=1}^{N}\text{Cost}_j^k(i))^2}$$

We then let

$$w_k(i) = \frac{1}{\text{Std}_k(i)}$$

where $w_k(i)$ is the weight given to property k of segment i. (If Std_k is 0, w_k becomes infinite; but this never happens in our experiments.)

Note that when we first establish the segmentwise correspondences between the signatures and the model, we do not yet know the weights that should be assigned to the properties. We therefore initially establish these correspondences using property weights that are all equal to 1. After these initial correspondences have been established, we can estimate the weights as described above, and then recompute the correspondences using these weights. Note that we have not normalized these weights so that they sum to 1 for each segment; if a segment is highly

64

segment	xpos	ypos	xspan	yspan	size	hist	enty
1	0.004	0.053	0.008	0.083	0.018	0.02	0.086
4	0.019	0.046	0.028	0.067	0.059	0.051	0.202
8	0.018	0.052	0.011	0.104	0.046	0.011	0.098

Table 4.1: Some standard deviations of matching costs (see Figure 4.16).

variable in all of its property values we want it to have low subtotal weight in computing the total cost of a match.

4.5 Experiments

Our experiment was performed on an 800-signature database, which was provided by Prof. Robert Sabourin of the École de Technologie Supérieure in Montreal, Canada. Sections 4.5.1 and 4.5.2 describe how the system was trained in preparation for these experiments.

4.5.1 Model construction and weight estimation

Using a mouse, we manually traced one hard-copy signature image for each writer to obtain the ordered sequence of points that constituted the model for that writer's signature. The signature image used for this purpose was chosen arbitrarily: it was the first signature of that writer in the database. To verify that the model is a good representative of the signatures of the author, we can compute the property differences between the model and the signatures. Ideally, the mean of these differences should be zero for each segment and each property. If the model is biased, the differences will be skewed; we did not find this to be the case.

We performed pointwise matching of every signature in the database to every model, as described in Section 4.3. This allowed us next to compare all the signatures to each model segmentwise, as described in Section 4.4. By doing this with a training set of genuine signatures, we were able to estimate the variability (i.e., the standard deviation) of each property of each segment of each writer's genuine signatures. (As described in Section 4.4.3, we did the matching initially using property weights of 1.) We took the reciprocals of these standard deviations as the weights to be given to the segment properties in the experiments.

4.5.2 Threshold selection

Using the reciprocal standard deviations as weights, we matched every genuine signature in the training set to the model for that writer's signature, and we also matched every other writer's signature in the database to the model for each writer. In this section we discuss the thresholds selected using the means and standard deviations of the costs of these matches (as in Chapter 3.6.2, Method I).

We assume that the match cost distribution of the kth training set of genuine signatures when matched to that writer's model is Gaussian with mean m_k and standard deviation σ_k. For the rest of the writers in the database, the cost distribution $g(x)$ of comparing their signatures to any individual writer's model will also be assumed to be Gaussian with mean M and standard deviation Σ. (For a more detailed discussion, please refer to Section 3.6.2.)

Figure 4.17(a) shows the cost distributions for each of 20 writers, matched against the first writer's model. (The horizontal axis represents the writers; the vertical axis represents the costs;
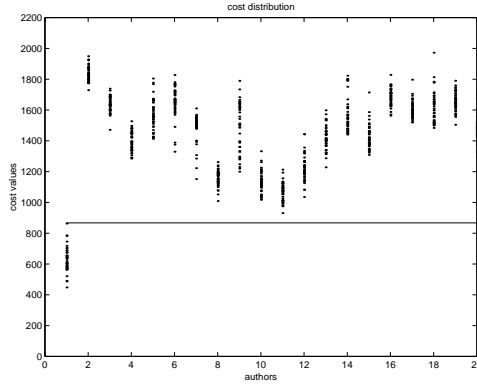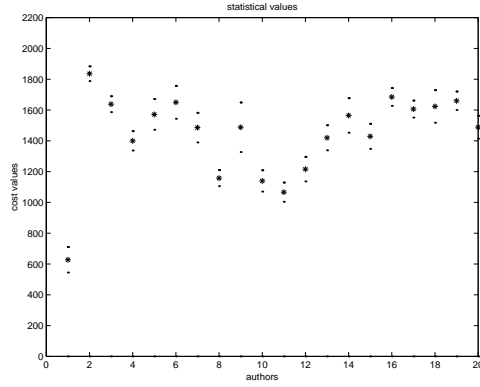
(a)   (b)
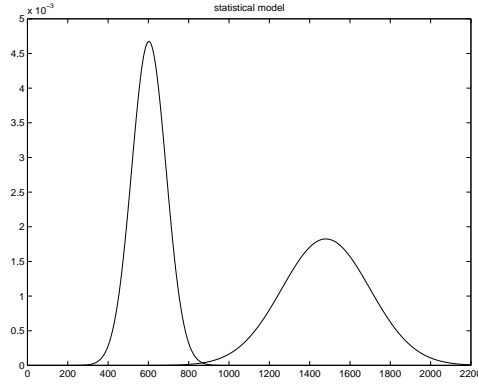
(c)

Figure 4.17: (a) The plotted cost of the data set vs. model A. (b) The mean and standard deviation of the training set. (c) The statistical model for the signatures and "forgeries" of writer A.

each dot is the cost of a particular comparison.) Figure 4.17(b) shows the means and standard deviations of these costs ($*$ = mean; dots = $\pm$ one standard deviation). Figure 4.17(c) shows the Gaussian distributions of the signatures and forgeries for the first writer's model—i.e., $f(x)$ and $g(x)$ defined in Section 3.6.2 .

$$-\frac{(x-M)^2}{2\Sigma^2} + \frac{(x-m_k)^2}{2\sigma_k^2} = \ln\frac{\Sigma}{\sigma_k}$$

The solution x of this equation is the optimum theoretical threshold. (In Figure 4.17(c), this threshold is at the point where the two curves cross.)

In Figures 4.18-4.20, we show cost distributions and Gaussian distributions for three other writers. Tables 4.2-4.5 give the values of the statistics for writers A–D, respectively, for training sets of sizes 10, 20, and 40.

66

training set size	10	20	40
M	1479.3	1391.9	1309.8
Σ	218.6447	202.8496	193.4844
m_k	601.3928	575.7037	564.9343
σ_k	85.369	67.7919	64.419
threshold	867.4316	798.0802	798.8174

Table 4.2: The values of the statistics for writer A.
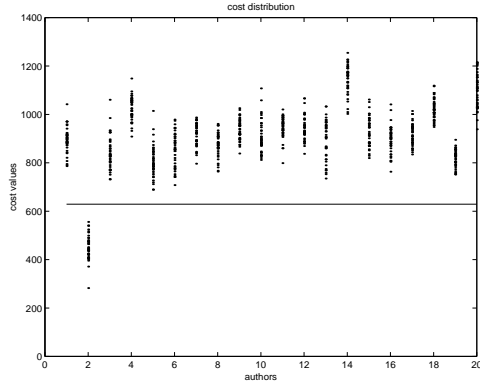
(a)
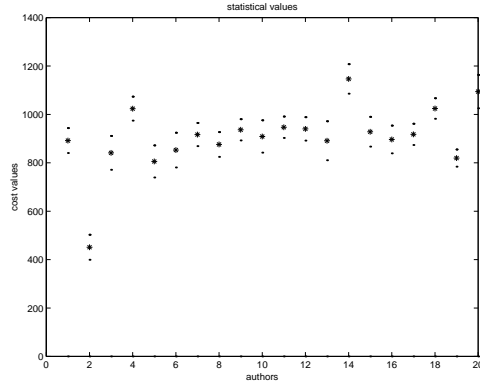
(b)
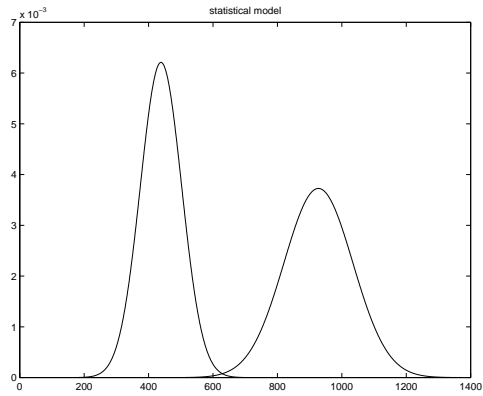
(c)

Figure 4.18: Analogous plots for writer B.

training set size	10	20	40
M	927.2542	866.7179	838.8479
Σ	107.0505	95.8272	92.8574
m_k	438.5639	412.7391	407.8121
σ_k	64.2343	57.7624	47.6639
threshold	628.9803	589.6011	560.7881

Table 4.3: The values of the statistics for writer B.
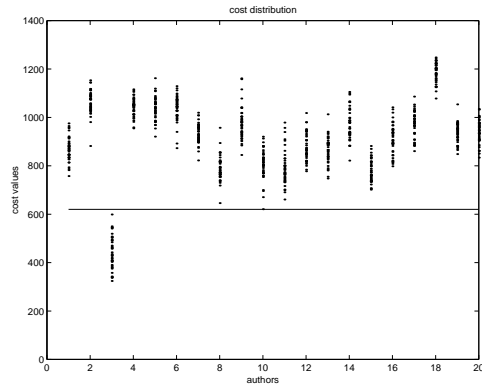
(a)
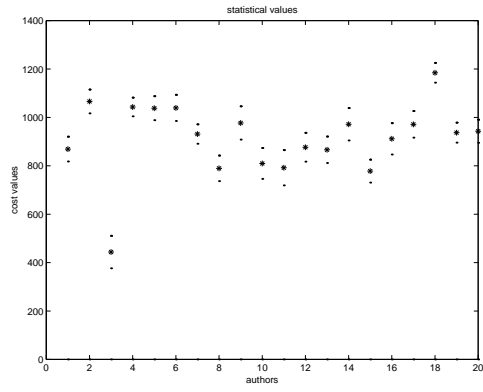
(b)
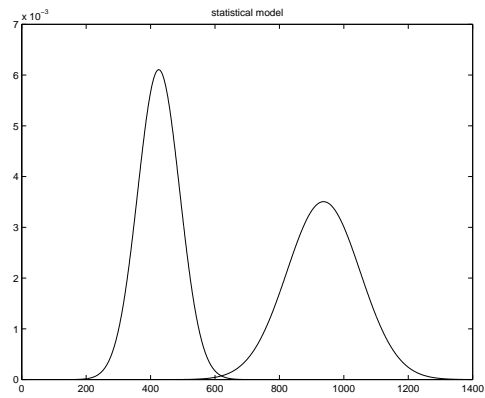
(c)

Figure 4.19: Analogous plots for writer C.

training set size	10	20	40
M	936.9539	900.4328	866.513
Σ	113.7943	106.792	106.1961
m_k	425.3815	415.3224	405.0194
σ_k	65.3352	59.229	59.7664
threshold	619.962	596.0017	579.0361

Table 4.4: The values of the statistics for writer C.
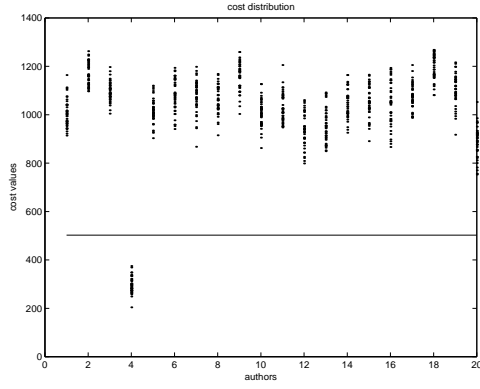
(a)
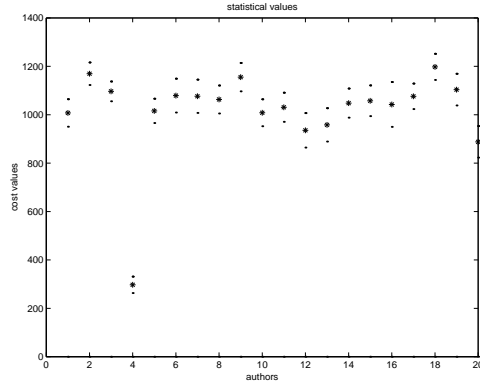
(b)
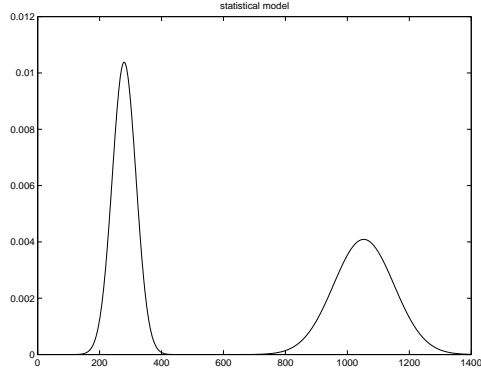
(c)

Figure 4.20: Analogous plots for writer D.

training set size	10	20	40
M	1052.9	1007.3	978.8713
Σ	97.5336	90.861	95.7654
m_k	279.2172	270.4626	272.3688
σ_k	38.4269	27.9040	30.7075
threshold	502.3599	447.6062	448.5446

Table 4.5: The values of the statistics for writer D.

	P_{FR}	P_{FA}
without training	0.53%	1.44%
training set 10	0.27%	0.40%
training set 20	0.13%	0.39%

Table 4.6: Error rate over the database.

4.5.3 Results

For each model, the cost was calculated for all 760 forgeries, as well as for the genuine signatures excluding the training set. The first row of Table 4.6 shows the false rejection rate P_{FR} and the false acceptance rate P_{FA} when the weights were all set to 1 and the threshold was chosen by hand to minimize $P_{FR} + P_{FA}$ for each writer. The second and third rows show the results when training sets of size 10 and 20 were used to determine the weights and to find theoretical thresholds by the method discussed in Sections 4.5.1-2. (We divided the 40 signatures of each writer into four groups of 10 each and two groups of 20 each; the numbers given in Table 4.6 are averages taken over the four 10's and the two 20's). We see that even using a small training set greatly reduces P_{FA}, and that using a larger training set keeps P_{FA} small while greatly reducing P_{FR}.

4.6 Discussion

We have described a method of performing model-guided tracing and segmentation of off-line signatures, thus allowing us to match the signatures to the models. We have shown that the match costs obtained in this way can be used to distinguish genuine signatures from random forgeries.

In our approach, the sizes of the questioned signatures need to be approximately the same as the size of the model because the model is not scale-invariant. They do not need to be exactly the same size, because the tracing process is elastic. If necessary, we could redefine the model to make it scale-invariant.

In pilot experiments not described in this chapter, we observed that real on-line models seem to work better than traced models. This may be due to irregularities in the traced models, due to unsteadiness of the tracing movement.

Our tracing process can also be used for handwritten character recognition, as illustrated by the single-letter examples shown in Section 4.4. The process can also be used to establish a local correspondence between a model and a questioned signature, so that information at the stroke level can be evaluated and used for skilled forgery detection; this application of the process will be investigated in the next chapter.

A significant feature of our approach is that we obtain not only a reasonably good segmentation but also a segment-wise correspondence between the model and the questioned signatures. This enables us to examine both global and local features of the questioned signature. Thus, it can serve as an initial step in the detection of skilled forgeries. The other important feature is that we are able to associate different weights with different features down to the stroke segment level. Thus, we are able to put more weight on the most consistent features. This makes our verification process closer to that used by expert document examiners who examine the most important and most invariant features of the questioned signature. Up to now, we have only discussed random forgeries. We have also done some experiments on simple forgeries.

Since our cost function was based on structural invariance, not on artistic qualities such as the evenness of the strokes, etc., our approach does not work well on skilled forgeries, as expected. The positions of the stroke segments make a difference in random forgeries since different names have different stroke components, so that the positions of the stroke segments are different. In some simple forgeries, the positions can be different due to the use of different personal touches in different letters. For skilled forgeries, however, the positions of the stroke segments can be exactly the same as in a genuine signature. The shapes and slopes are different for random and simple forgeries because different people write curves and loops and connections between letters differently. In skilled forgeries, these features can be studied carefully and forged accurately. The relative sizes of the stroke segments are useful for detecting random forgeries because this feature can distinguish between the stroke segments of letters of different sizes such as "e" and "l". This feature may also be useful in detecting some simple forgeries in cases where people write letters in their signatures unconventionally but this fact is unknown to the forger. Thus our features are useful for detecting random and simple forgeries, but they fail on skilled forgeries.

Chapter 5

Skilled Forgery Detection Using Local Analysis

5.1 Introduction

Research has been very active in the field of forgery detection. However, most of the work has been on random forgery detection, partly because random forgeries are a large fraction of forgery cases [7], and partly because of the difficulties encountered in analyzing detailed information embedded at the stroke level, which must be done in order to detect skilled forgeries.

Applications of random forgery detection include, for example, check clearing and credit card validation. Even with genuine signatures shown on the backs of credit cards, people often fail to compare them. If automatic signature verification can be achieved, it can help prevent fraud in financial transactions which cost retailers hundreds of thousands of dollars each year. Such a system should be able to extract useful features from a questioned signature, compare them with an existing model, and make a decision as to whether the signature is genuine or forged. In most applications a low false alarm rate is required, as false alarms need additional processing, and in the worst case, may have legal consequences.

The applications of skilled forgery detection are somewhat different from those of random or simple forgery detection. Instead of aiming at mass verification, skilled forgery detection usually targets small volumes of data. One common application would be in the forensic sciences which generally rely on expert document examiners. Due to the subtleness of detecting skilled forgeries and the nature of the applications, the aim of automatic skilled forgery detection can be providing an aid to the document examiner. Such a system could provide tools for the document examiner to quantitatively measure features of a questioned signature and compare them with features of known signatures. Nevertheless, automatic skilled forgery detection remains an elusive challenge.

Unlike random and simple forgeries, skilled forgeries need to be examined carefully even by experienced document examiners. Such forgeries may have poor line quality, resulting from properties of handwriting related to movement, including speed, continuity and uniformity, pen pressure, freedom or hesitation, rhythm and writing skill [7]. Unfortunately however, skilled forgers attempt to mimic the style of the author; hence skilled forgeries can be difficult to detect. On the other hand, due to the fact that the forger has to concentrate on mimicking the style of the author, skilled forgeries are very difficult to do ballistically; thus skilled forgery usually affects the rhythm and uniformity of writing. The result of lack of rhythm and uniformity contributes to the fact that forgeries have poor line quality.

Skilled forgeries are often subclassified into traced and simulated forgeries. Such forgeries are drawn, not written; thus they do not have the normal changes that occur along smooth strokes. As we shall see in Section 5.3.2, a traced forgery displays even poorer line quality than a forgery done by simulation. In addition, both simulated and traced forgeries tend to

exaggerate the special features in a signature. For example, an extended cross stroke of a "t" will be exaggerated and made even more prominent in a simulated signature. These special features or personal touches are places that an expert document examiner pays attention to. At the same time, forgers often neglect inconspicuous structural details, subtle shading, and writing direction. These subtle differences between genuine signatures and skilled forgeries have to be compared locally on the stroke level.

In Chapters 3 and 4 (see also [48]), we described algorithms that segment a questioned signature and make a correspondence between the signature and a model. This makes it possible to examine the questioned signature on the stroke level. The work presented here is based on this correspondence.

5.2   Properties of a Stroke

Different writing instruments can have different effects on the appearance of a signature. Some writing instruments do not have much variation from one stroke to another. Widening, for example, is the major effect of pressing down harder when using a felt tip pen. For ball point pens, pressing down harder produces strokes that are both darker and wider. Both of these instruments favor width variation over darkness variation, while a pencil and a fine tip pen favor darkness variation.

Pressure changes from pressing the pen down on the paper occur in upstrokes and downstrokes. When comparing upstrokes and downstrokes over the entire signature, a lack of variation in width or darkness, coupled with the ability of the human observer to determine the writing instrument, can be an indication of possible forgery even if a model is not available. In our work presented here, we do not attempt to identify the writing instrument before further processing. However, if we could identify the type of pen used in writing the questioned signature, it definitely would be an advantage in detecting forgeries.

In this section, we discuss some of the differences between strokes that result from their being written by different authors as well as in different writing environments.

5.2.1   Differences in gradient magnitude between the edges of a stroke

The three fingers that typically hold the writing instrument exert forces at three different positions [14]. The force that presses down on the surface of the paper is mainly exerted by the index finger. The position and magnitude of this pressure on the writing instrument varies with personal habit. Figure 5.1 illustrates writing with an instrument that has a rigid tip while Figure 5.2 illustrates writing with a ball-point pen. The forces on a rigid-tip pen or on a pencil are directly transmitted to the paper, while the forces on a ball-point pen are transmitted to the paper through the ball and socket.

In Figure 5.1, a and b are the two boundary points when using a rigid tip pen. On the paper, these generate the two edges of the strokes. L_1 is the distance between a and the center of force, i.e., the position that fingers grab the pen while L_2 is the distance between b and the center of force. In Figure 5.2, A_1 and A_2 are two sides of the socket that the ball is in. In the rigid-tip case the tip is flat and L_1 and L_2 are different. Thus, the values of the friction balance at points a and b, the edges of the stroke, are different. The situation is more complicated with a ball-point pen. The ball is able to move in its socket. Thus when it contacts the paper, an area instead of an ideal point is in contact. The size of this area depends on the pressure used in pressing the pen down on the paper as well as the ball size. The smaller the angle α is, the closer the area is to one side of the socket edge. When pressing down, the ball is forced from

to center of force

L1   L2

a   b

enlargement of pen tip area

Center of force
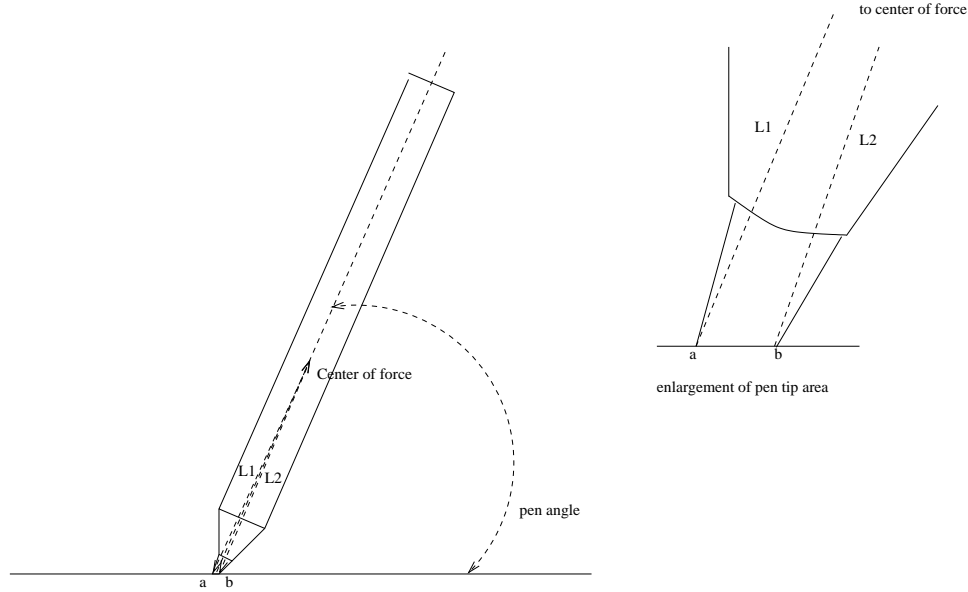
L1 L2

pen angle

a   b

Figure 5.1: Illustration of using a pencil.

A_1 to A_2 leaving a bigger gap at A_1 and a smaller gap at A_2. This makes the ink jet out more in the area around A_1 than in the area around A_2. If the pen is perpendicular to the paper, the ball is pushed upward, so that the ink should be more evenly distributed and less different on the two edges. Changing the force of pressing down also affects how much ink comes out. From these remarks we can see why the position of the grasp on the pen and the angle of the pen result in differences between the two edges along a stroke.

Figure 5.3 shows edge profiles of two horizontal strokes written by different people. Figure 5.3(b) shows a bigger difference between the two edges than Figure 5.3(a). In the figure, the horizontal axis shows the sample points, i.e., the consecutive points along the horizontal stroke. The vertical axis shows the gradient magnitude values at the points. The differences in Figures 5.3(a) and (b) show that due to difference in writing habits such as grasp position and pen angle, the gradient magnitude can be different on the upper and lower edges of a horizontal stroke; but this difference can vary from one writer to another. When the writer of Figure 5.3(b) made the horizontal stroke, the pen was less perpendicular to the paper than it was for the writer of Figure 5.3(a). This affects the differences between the gradient magnitudes on the upper and lower edges of the strokes.

The way a person holds a pen also changes as he make lines at different angles, or circles or loops. For example, we tend to hold the pen more nearly perpendicular to the paper when we make a vertical line than when we make a horizontal line. Figure 5.4 shows edge profiles of vertical lines written by the same people as in Figure 5.3. We see from the Figures that the difference in gradient magnitude on the two edges of a vertical line is smaller than that for a horizontal line. In the case of a writing instrument with a rigid tip, the difference between L_1 and L_2 (see Figure 5.1) gets smaller when the pen is more perpendicular to the paper as we write a vertical line. In the case of a ball-point pen, the difference in the gaps at A_1 and A_2 (see Figure 5.2) gets smaller as well. As a result, the difference in the forces on the two edges gets smaller. This also illustrates the fact that the significance of a feature measurement can differ from one stroke to another. In this example, the difference between the two sets of edges is greater for the horizontal lines. Evidently, the situation for a signature, where upstrokes and

74
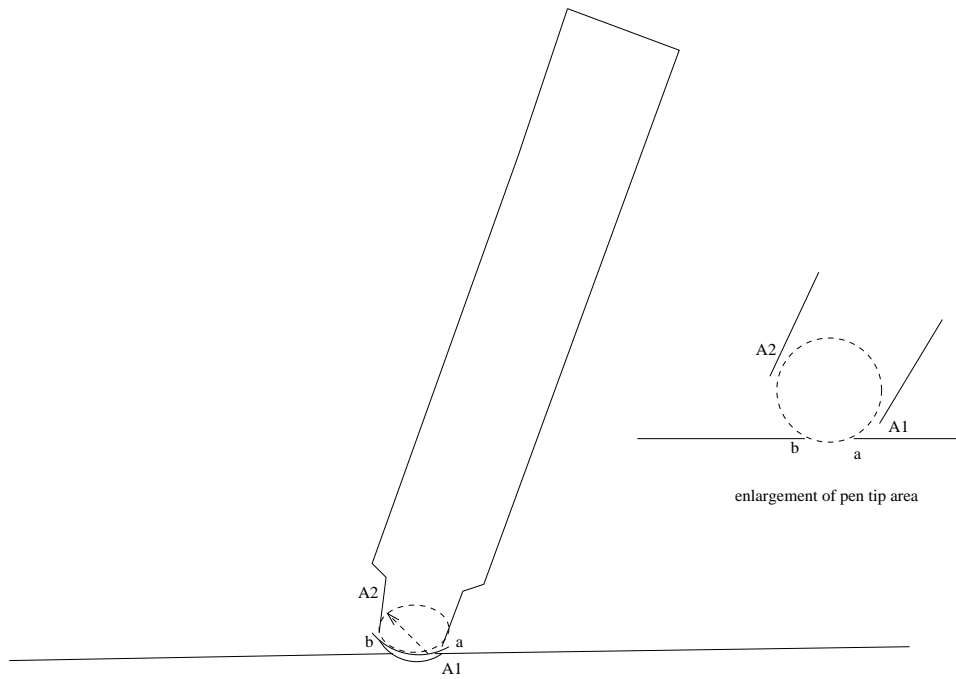
Figure 5.2: Illustration of using a ball-point pen.
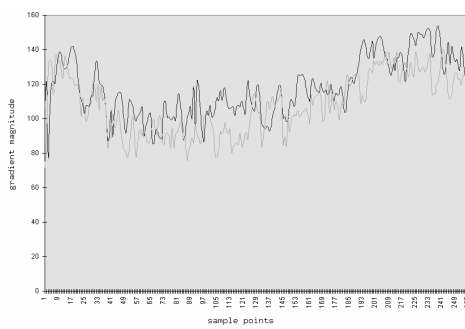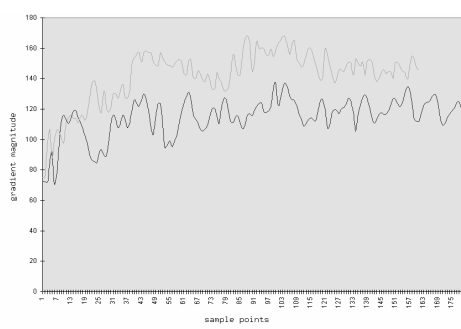
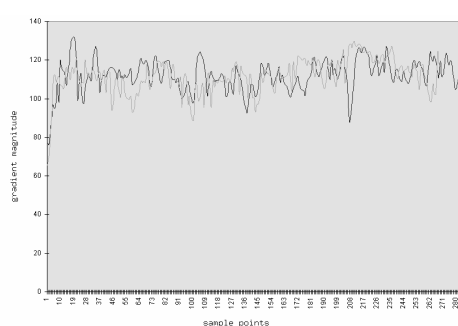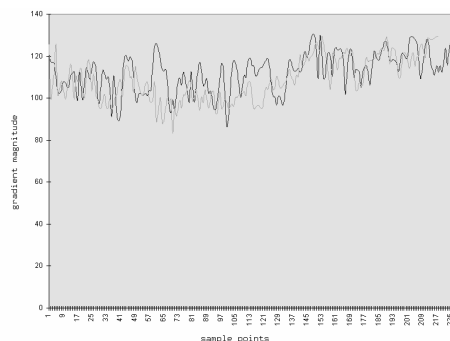(a)   (b)

Figure 5.3: The edge profiles of two horizontal lines.

<div align="center">(a)</div>   <div align="center">(b)</div>

<div align="center">Figure 5.4: The edge profiles of two vertical lines.</div>

downstrokes are made ballistically, will be more complicated than it is for horizontal and vertical lines, but it is clear that different writing habits generate strokes with different properties. We should also point out that left-handed people grasp writing instruments differently than right-handed people; we can therefore expect that the stroke-level properties in their handwriting are different as well.

5.2.2 Changes in grey level

The pressure on the pen tip varies as a stroke is generated. For example, horizontal strokes usually taper off. A person who writes normally does it ballistically. Changes in pressure are continuous and occur in a unique rhythmic fashion. The changes in pressure are also related to the velocity of the writing.

Figure 5.5(a) shows a naturally written upward loop that was captured using a pressure-sensitive tablet. The points shown in the figure were sampled at equal time intervals. Thus, the less dense the points are, the slower the writing is. Figure 5.5(b) shows the pressure changes in that loop (the key points are marked in both Figure 5.5(a) and Figure 5.5(b)); it shows a smooth decrease in pressure around the top of the loop. We see that the pen slows down toward the top of the loop; immediately after the loop, the speed picks up again and the pressure decreases. After the point labeled "d", the speed slows down and the pressure starts increasing again.

In a forgery, the forger typically writes with great precision, so the pen pressure changes less, or in an unconventional way. Figure 5.6 shows the pressure changes when tracing an upward loop. Tracing is also slower than writing; there are far more points in Figure 5.6(a), which was captured by the tablet. The horizontal axes of Figure 5.5 and Figure 5.6 are the sample points, and the vertical axes are the pressure values. In off-line writing, the curves may not be as smooth as these, because they will be affected by digital noise, as in Figure 5.7.

Change of speed affects not only the smoothness and the stroke width along the stroke, but also the friction balance at the pen tip. The pressure changes along a stroke therefore affect the gradient magnitudes along the edges of the stroke, as well as the gray level changes along the stroke. In Figure 5.7 and Figure 5.8 we show examples of gradient magnitude and gray

<div align="center">76</div>
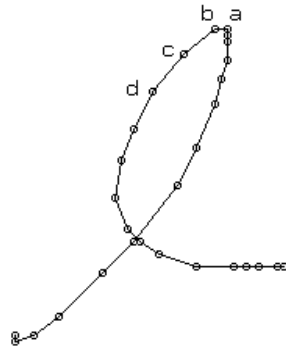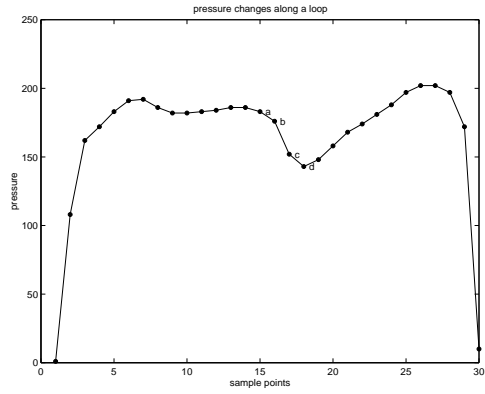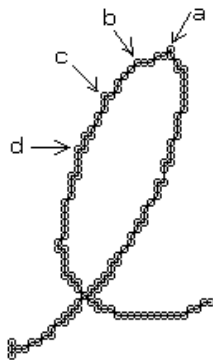
(a)   (b)
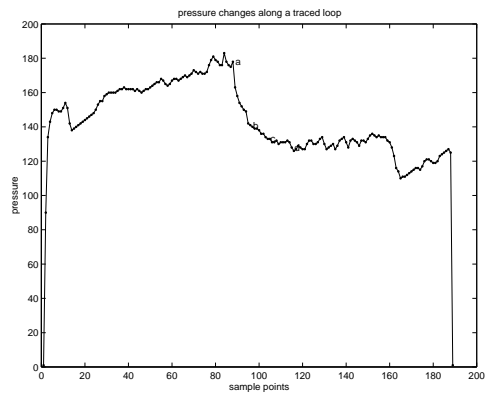
Figure 5.5: Pressure changes in writing an upward loop: (a) The loop. (b) The pressure profile.

(a)   (b)

Figure 5.6: Pressure changes in tracing an upward loop: (a) The loop. (b) The pressure profile.
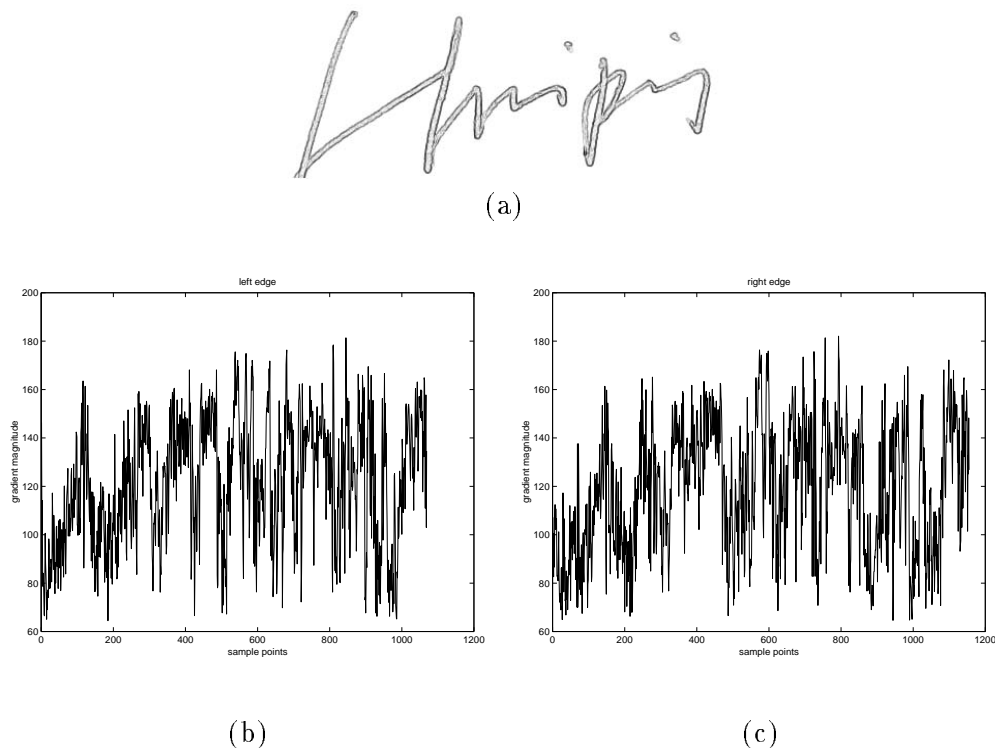
(a)

(b)   (c)

Figure 5.7: An example of gradient magnitude changes in an off-line signature: (a) The signature. (b) The left edge profile. (c) The right edge profile.

level changes along strokes. We see from the figures that downstrokes are generally darker than upstrokes. Figure 5.7(b) shows the gradient magnitude of the left edge (as defined in Section 5.3) of Figure 5.7(a), while Figure 5.7(c) shows the gradient magnitude on the right edge. Even though the profiles are affected by noise, we can see that the low frequency components, which represent the gradient magnitude "flow" along the strokes, do change from sub-stroke to sub-stroke.

5.2.3   Changes in stroke width

Since the pressure put on the pen tip changes along a stroke, the width of the stroke changes with it. For example, the pressure in the upward part of the stroke in Figure 5.5 is smaller than that in the downward part of the stroke. This makes the width of the stroke in the upward section smaller than that in the downward section. The stroke width changes when people write naturally. In Figure 5.8 we see that the stroke width changes along the strokes; for example, the downstrokes are wider than the upstrokes. When trying to forge a signature, however, due to the careful study of the signature or the careful action involved in tracing the signature, the pressure is more evenly distributed along the stroke, making the stroke width more uniform.

5.2.4   Stroke smoothness

Simulated and traced forgeries are drawn, not written. Thus they do not have all the natural changes along the strokes. A traced forgery displays poorer line quality than a forgery by

78
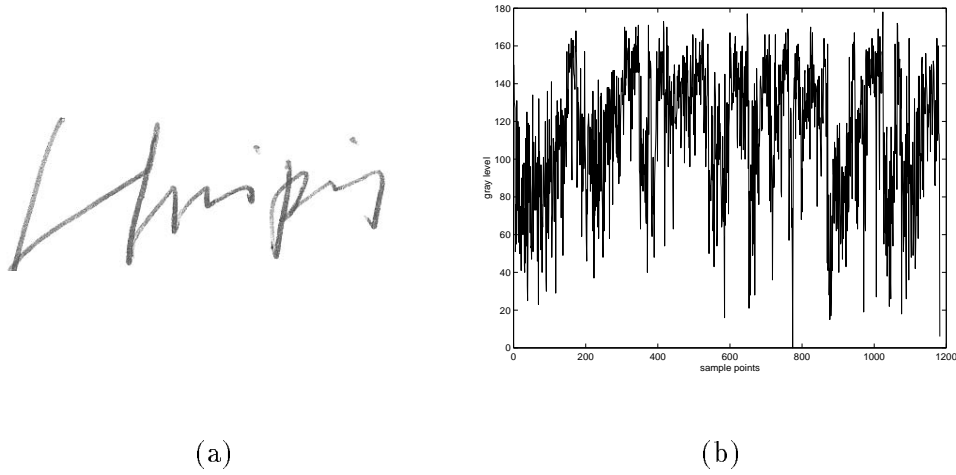
Figure 5.8: An example of gray level changes in an off-line signature: (a) The signature. (b) The gray level profile.

simulation. In one experiment [12, 13], over 97% of the traced signatures contained strokes with obvious tremor. Forgers tend to put together a signature letter by letter, stroke by stroke. Some parts of the signature are touched up or reworked. Thus, the strokes of genuine signatures are smoother than those of forged ones. However, due to the digitization of the signature images and to noise introduced in the scanning process, this smoothness is hard to measure.

5.3 Our Approach

In the previous chapter, we described a model-based algorithm for segmenting a questioned signature. During the process of segmentation, a stroke-segment-wise correspondence between the model and questioned signature is established. The work on skilled forgery detection described below is based on such a correspondence. Unlike approaches that use global statistics of the local properties, the stroke-segment-wise correspondence enables us to compare the questioned signature with the model on the stroke level.

It is also important to note that by using the differences between model features and questioned signature features, we can adapt to deal with writers who naturally write slowly and meticulously, or writers who may have tremor due to illness or age.

Because variations in people's writing result in pressure changes which are reflected in gray level changes, as discussed in Section 5.2, we first obtain the gray level gradient magnitude along both edges of a stroke as well as the grey level value along the stroke. These values are compared with those of the model stroke by correlation. The details of this process are discussed in the following subsections. We assume that we have established a strokewise correspondence between the questioned signature and the model, as described in Chapter 4, and have eliminated random forgeries.
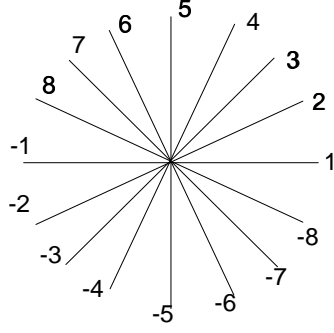
Figure 5.9: Definition of stroke directions.

5.3.1 Finding the edges

To obtain the edges of the stroke, we search along the direction perpendicular to the stroke direction (Figure 5.9). The search process is illustrated in Figure 5.10. A stroke that is traced under the guidance of a signature model is usually within the thickness of a signature stroke, but it does not necessarily form a smooth segment. The edge segments obtained from model-traced segments therefore need to be smoothed; otherwise, situations like that shown in Figure 5.11 can arise, where the points occur in the wrong order. The edge points usually form smooth curves, which can be obtained by tracing the edge points based on 8-connectivity. However, the contour segments obtained by tracing the edge points in an edge image contain no logical information about the signature, meaning that they do not necessarily start at the correct point, nor do they follow the direction defined by the model. Thus, we need additional processing to make the contour segments into smooth edge loci. A detailed discussion of these processes follows.

The contour edges contain edge points on both sides of the strokes. When moving along a stroke, the contour edge points on the left form a curved line, defined here to be the *left edge* of the stroke, while the contour edge points on the right form a curved line, defined here to be the *right edge*. For forgery detection purposes, we need to identify the left edge and right edge of each model-traced stroke.

The stroke direction

In this section, we define the stroke direction for the purpose of obtaining the initial ordered edge points illustrated in Figure 5.11. First, each traced stroke point is compared with its previous point and a stroke direction is calculated for each point. The stroke direction depends on the position of the current point relative to the previous point. The definitions of the directions were shown in Figure 5.9. Each quadrant is divided into four angular regions, each representing an angular range of 22.5^o. For each pair (current point p, previous point p_o), dx and dy are defined by

$$dx = p_x - (p_o)_x$$

$$dy = p_y - (p_o)_y$$

A direction value is assigned to each stroke point depending upon which range its dx and dy fall into. If the current point is the same as the previous point, i.e., $dx = dy = 0$, the direction at the point is 0.
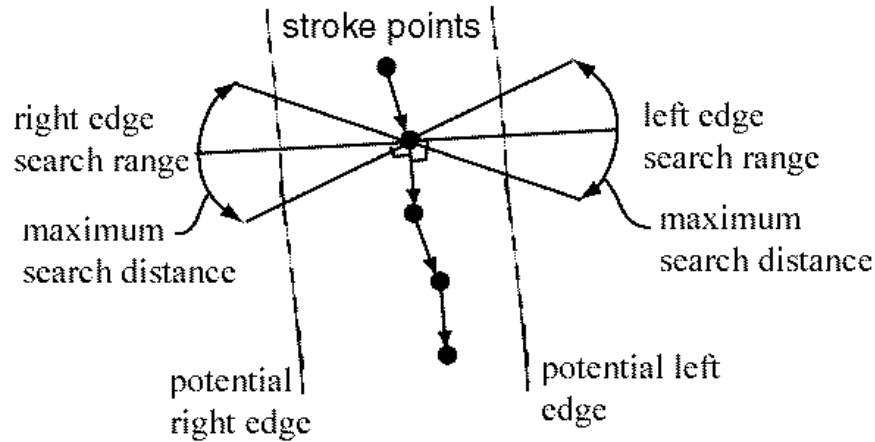
Figure 5.10: Illustration of edge search range.

The search range

In this section, we define the search range for each of the stroke points that we use to find the corresponding edge point. When each point on a stroke is examined, search ranges for the left edge and right edge are constructed according to the direction at the point. The search range for the left edge covers three direction segments centered at 90° counter-clockwise from the current stroke direction. Similarly, the search range for the right edge covers three direction segments centered at 90° clockwise from the current stroke direction.

The stroke edges

For each stroke point, a search for its left contour edge is first conducted within the left search range just defined. This search range contains three directions. Starting from the center direction, each point along the current search direction from the current stroke point is examined in the contour edge file to see if it is a contour edge. If a contour edge point is found before the maximum distance is reached, a left edge point has been found and the search for a right edge point begins. If no contour edge points are found in the current search direction, the other two directions are searched in the same manner. If no contour edge points are found in the entire search range, the search for a left edge point has failed for that point in the stroke segment.

Right edge points are similarly searched for in the right edge search range, and are considered to be found if a contour edge point falls within one of the search ranges. This search for left and right edge points is performed for every traced stroke point.

Edge post-processing

Using the above procedures, all the left edge points are grouped into left edge segments according to the order of their associated stroke points, and all right edge points are grouped into right edge segments in the same order. However, because the stroke segments may not be smooth, the left and right edges may not be ordered. For instance, as illustrated in Figure 5.11, two left edge points may be in reverse order from the direction of the stroke due to a zig-zag pattern in the stroke segment. Therefore, a post-processing procedure for the edge points is implemented.
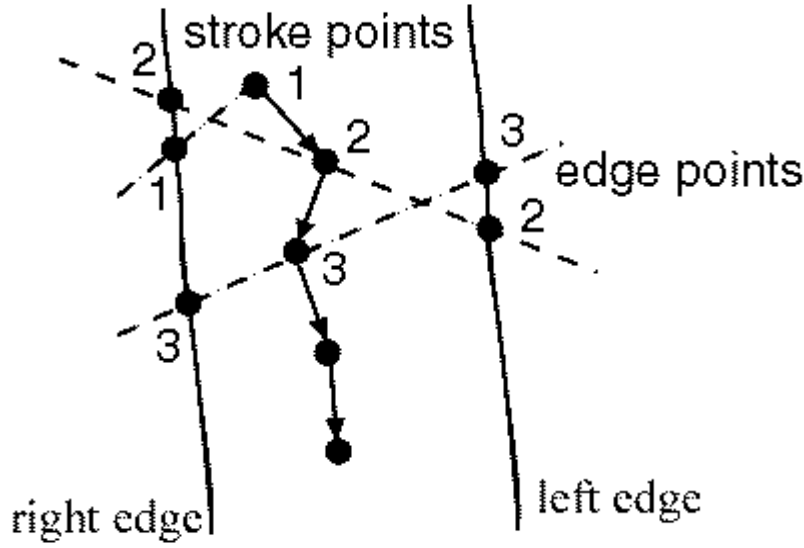
81

Figure 5.11: Illustration of reversed edge points.

Since one contour segment may contain the left/right edge for different edge segments, the segment number of the contour segment and the segment number for the edge segment may be different. We denote the contour segment number by i and the index number, i.e., the point number on that contour, by p. When a left or right edge point is found on an edge contour segment, the contour segment number i on which the edge point is found, and its index p on this contour segment, are recorded. When we obtain the corresponding left and right edge points of the stroke segment as in Section 5.3.1, we obtain a set of labels(i, p). We then examine these ordered sets. From these ordered sets, we can copy a section of the edge contour into sections of the edge segments. We break up a contour segment into sections of an edge segment if any of the following conditions is met:

- The associated model-traced stroke breaks, meaning that the segment ends. This is illustrated in Figure 5.12. In the Figure, the contour continues to wrap around the other side of the edge after the stroke finishes.

- The contour segment number changes from that of the previous point, meaning that the contour breaks, which could be caused by noise on the edge, and the stroke segment should continue as in Figure 5.13.

- The contour segment number is the same as that of the previous point, but the index number changes significantly from that of the previous point, i.e. the change exceeds a pre-determined threshold. This can happen when an edge point is accidentally found on the wrong edge of a nearby stroke segment.

For each edge segment, the minimum index number p_{min} and maximum index number p_{max} on the same contour segment are obtained. Instead of grouping these points into a section of the edge segment, the portion of contour segment i between points p_{min} and p_{max} is copied to avoid situations such as that shown in Figure 5.11. Instead of using (1 2 3) as the right edge, (2 1 3) is used. This new edge segment is guaranteed to be smooth and continuous.

82

Figure 5.12: Illustration of a stroke ending while the contour does not.

Figure 5.13: Illustration of contour segment breaking while stroke continues.

5.3.2 Cost computation

Figure 5.14(a) shows an example of a genuine signature; its left and right edge gradient profiles are shown in Figures 5.14(b) and (c). Figure 5.15(a) shows an example of a simulated forgery. Its edge gradient profiles are shown in Figure 5.15(b) and (c). In Figure 5.16(a), a traced forgery is shown and the corresponding edge gradient profiles are shown in Figures 5.16(b) and (c).

As discussed in Section 5.2, genuine signatures are generated with a smooth rhythm. As a result, the gradient magnitude changes smoothly yet rhythmically along the edges of each stroke. With forgeries, however, less gradual changes may be observed. The edge envelope of the profile for the genuine signature has more rhythmic changes than do the other two envelopes. Of the three, the traced forgery has more carefully drawn features; thus its edge profile has the smallest changes, as we see by looking at the upper portion of the envelope of the profile. The simulated forgery is done more naturally, which can be detected in the profile. However, it differs from the genuine signature as a result of the writer's different writing habits.

In particular we see that the fluctuations in the profile are denser for the forgeries, especialy the traced forgery, meaning that the forgeries have more high-frequency components, possibly because of tremor, hesitation, and other non-ballistic factors. The profile of the genuine signature has a rhythmic low-frequency component; the simulated forgery has another form of low frequency component, and is more rhythmic than the traced forgery. Among the three, the low-frequency component in the traced forgery is the most flat, having less change in magnitude along the edges; this shows that traced forgery has the poorest line quality and involves more drawing than ballistic writing. The differences between the left and right edges are different in the three examples because of the different writing habits of different people; these differences are subtle and hard to measure.

Correlation is often used to compare signals that are shifted in time, but the signals generated in different samples of writing are not exactly shifted versions of each other, even for genuine signatures. The signals deform at various positions and times. Additionally, the signals are affected by digitizing noise, etc.

To better compare the model and the questioned signature, we compute average feature values for each stroke segment. We use the gradient magnitude feature as an example. As in [48], we segment the strokes at the local vertical minima and maxima. Averaging the gradient magnitude in each stroke segment greatly reduces the effect of noise. As we have pointed out, the force on the paper for upward and downward strokes should be different. Thus, when we average the gradient magnitude, the averages that we obtain from upward and downward strokes should be quite different. It should be recognized, however, that some people have a bigger difference in the upward and downward strokes while others have smaller differences.

Figure 5.17 and Figure 5.18 show the left and right edge gradient magnitude features for a genuine signature, a simulated forgery, and a traced forgery after averaging over each stroke segment. In this example, we see that the genuine writer writes rhythmically, with a careless smooth ending. The simulated forgery follows another type of rhythm that possibly characterizes the forger. The traced forgery has more ups and downs indicating possible tremor; as we have discussed, tracing is a very unnatural type of writing. Additionally, because of the fact that forging goes against one's normal style of writing, the ups and downs of the average gradient magnitude in each stroke segment are more different between the left and right edges. Although the magnitudes differ in the left and right edges of a genuine signature, they more or less follow the same pattern.
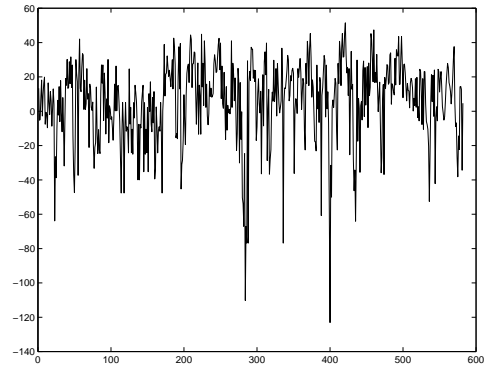
Let $E_m(i,j)$, $j = 0, 1, \ldots, M - 1$ denote the gradient magnitude along one of the edges of the model stroke segment i. Let $E_s(i,j)$, $j = 0, 1, \ldots, N - 1$ denote the gradient magnitude
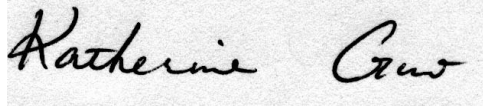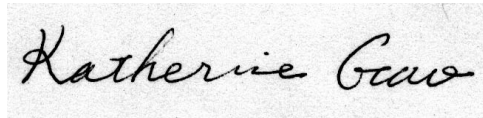
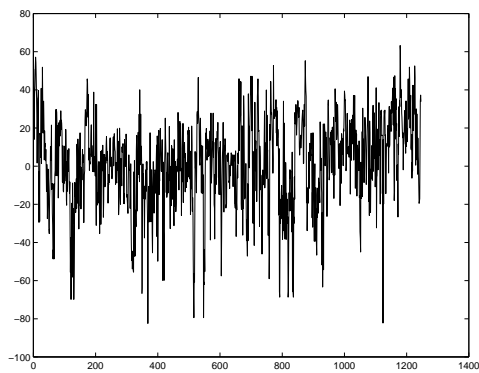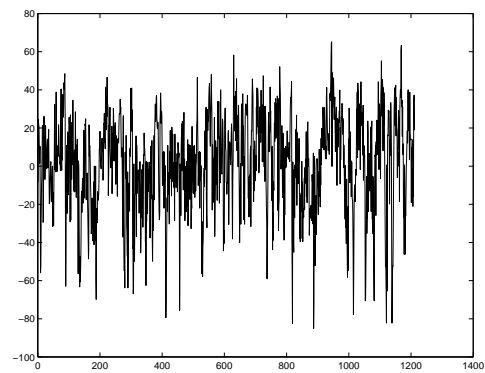(a)
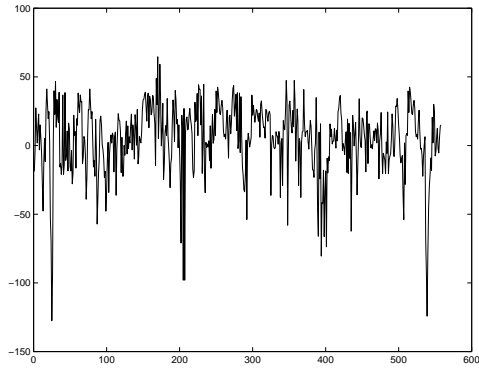
(b)   (c)

Figure 5.14: (a) Example of a genuine signature. (b) The left edge gradient profile of the signature. (c) Its right edge gradient profile.
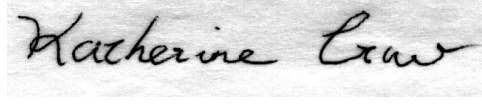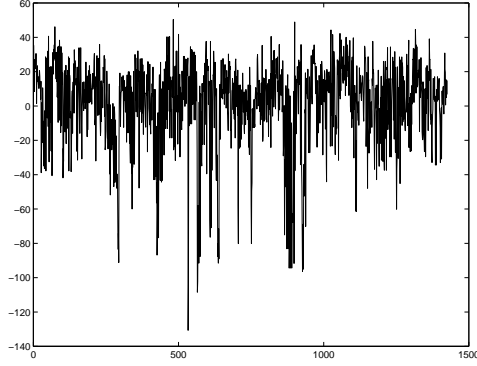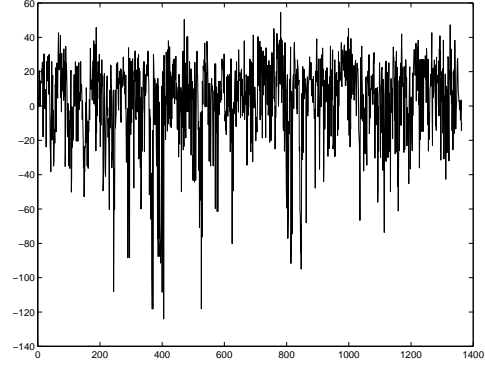

(a)

(b)   (c)

Figure 5.15: (a) Example of a simulated forgery. (b)The left edge gradient profile of the forgery. (c) Its right edge gradient profile.

(a)

(b)   (c)

Figure 5.16: (a) Example of a traced forgery. (b)The left edge gradient profile of the forgery. (c) Its right edge gradient profile.

along the corresponding edge of the corresponding signature stroke segment. Then

$$F_m(i) = \frac{1}{M-1} \sum_{j=0}^{M-1} E_m(i,j)$$

$$F_s(i) = \frac{1}{N-1} \sum_{j=0}^{N-1} E_s(i,j)$$

are the average of the gradient magnitude of segment i for the model and signature, respectively.

To compare the gradient magnitude of the questioned signature with that of the model, we compute

$$C_{mag} = \sum_i (F_m(i) - F_s(i)) * (F_m(i) - F_s(i))$$

for both edges.

Similarly, we compare the gray levels by computing the sum of the square differences of the average gray levels on the stroke segments for the model and the questioned signature. The same definition is used to compare the stroke widths.

To compare the gradient directions we proceed as follows. We first define a 16-component vector, where the components are the numbers of steps in the segment, from sample point to sample point, in each of the 16 directions illustrated in Figure 5.9. These numbers are normalized by the total number of sample points in the segment, so that each component of the vector represents the fraction of moves that the segment makes in a specific direction. The difference between the two vectors is defined as the sum of the squares of the differences between
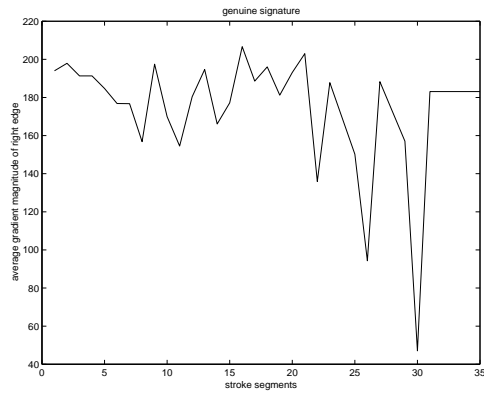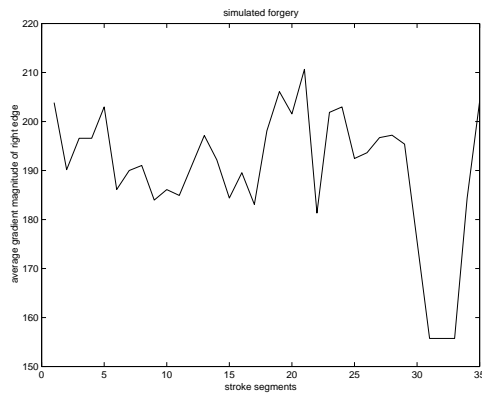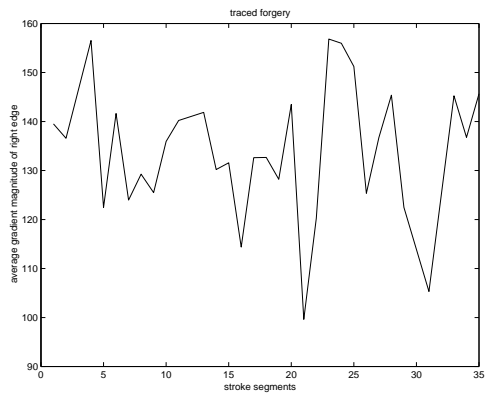
86
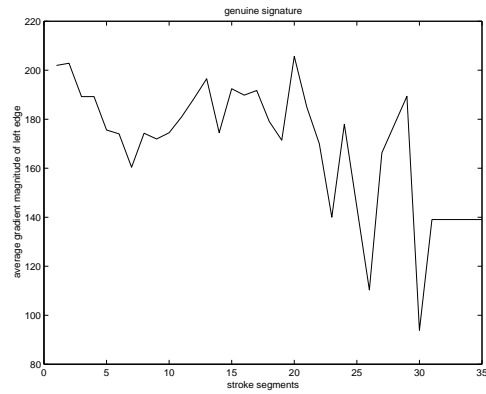
(a)
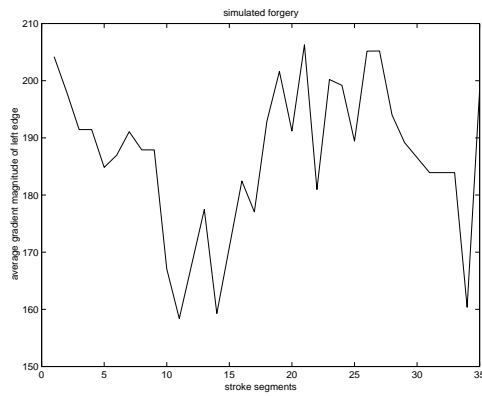
(b)
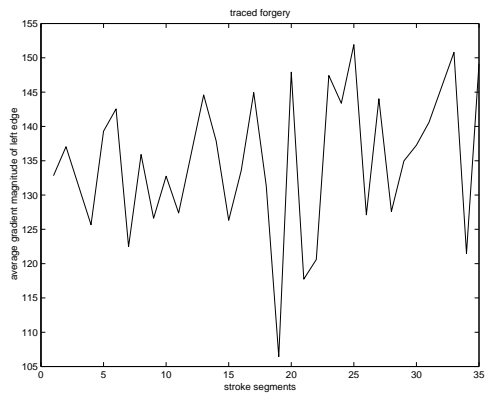
(c)

Figure 5.17: The left edge gradient profile after taking the mean of each segment: (a) A genuine signature. (b) A simulated forgery. (c) A traced forgery.

(a)

(b)

(c)

Figure 5.18: The right edge gradient profile after taking the mean of each segment: (a) A genuine signature. (b) A simulated forgery. (c) A traced forgery.

writer	forged by
A	other, F
B	D (blue ink), E
C (blue ink)	other, other
D	C, H
E	D, F
F	B, E
G	other, J
H (blue ink)	D (blue ink), G
I	other, other
J	B, D

Table 5.1: Summary of generation of the database.

corresponding components. We compare these gradient directions by computing the sum of the squares of the differences between corresponding components in the model and the questioned signature.

5.4 Experiments

5.4.1 Data collection

We studied the differences between genuine signatures, simple forgeries, simulation forgeries and traced forgeries using a database of 350 signatures that we collected. There were ten authors, with five genuine signatures, ten simple forgeries, ten simulated forgeries and ten traced forgeries for each author.

The ten authors each generated five genuine signatures. Each author's signature was then forged by two of the other writers. Each person who forged an author's signature did five simple forgeries, in which the person simply signed the name using his/her own style. The person also tried to simulate the genuine signature five times, after being given time to practice as many times as desired. Tracing paper was then provided, and each person traced the five genuine signatures. We did not put any restriction on the writing instruments. However, all the writers used ball-point pens with various tip thicknesses. Table 5.1 shows the authors and their forgers. Unless otherwise specified, the signature or forgery was in black ink. In the table, "other" refers to a person whose signature was not in the database.

5.4.2 Results

Simple forgeries can be easily detected using the structural cost functions we used in Chapter 4 and [48] for random forgery detection, as illustrated in Figure 5.19. [In Figure 5.19, the horizontal axis is the samples, where the first five are genuine signatures while the last ten are simple forgeries.] The genuine signatures are labeled "g" while the simple forgeries are labeled "s". The overall result on the simple forgery detection is 1/50, i.e., 2% for false rejection and 5/150, i.e., 3.33% for false acceptance.

For each model, edge profiles are obtained using the method described in Section 5.3 based on the edge information and the model strokes. Specifically, after segmenting the questioned signature into stroke segments based on the model, the edge profiles are obtained for each stroke
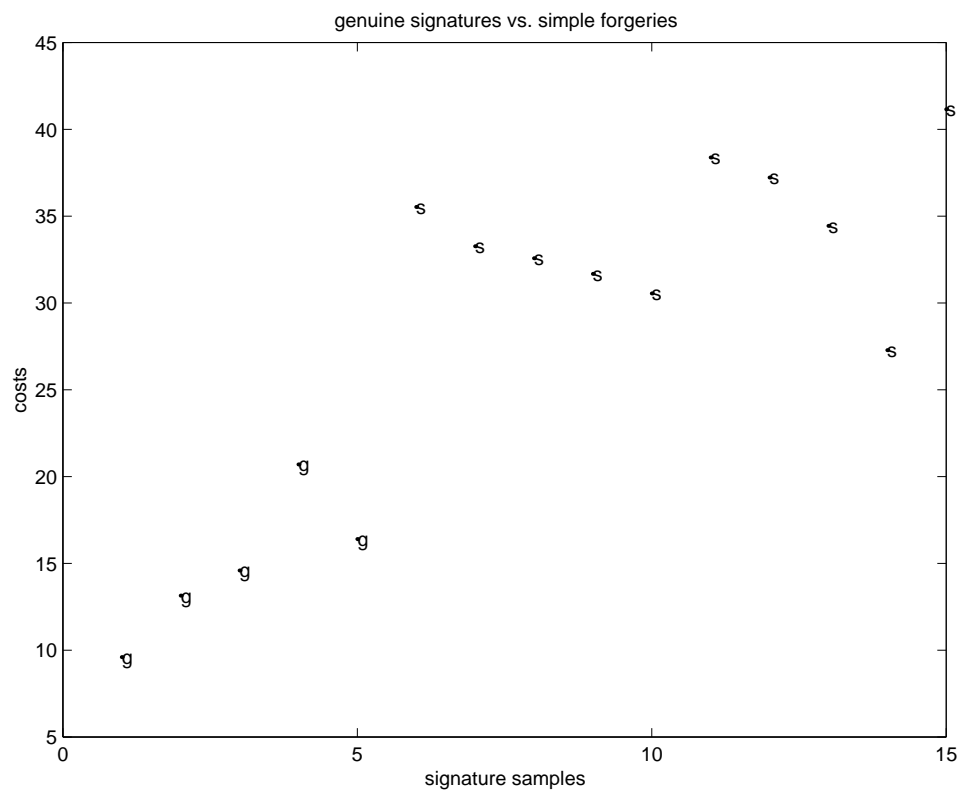
Figure 5.19: The costs of five genuine signatures (1–5 on the x-axis) and ten simple forgeries (6–15 on the x-axis).
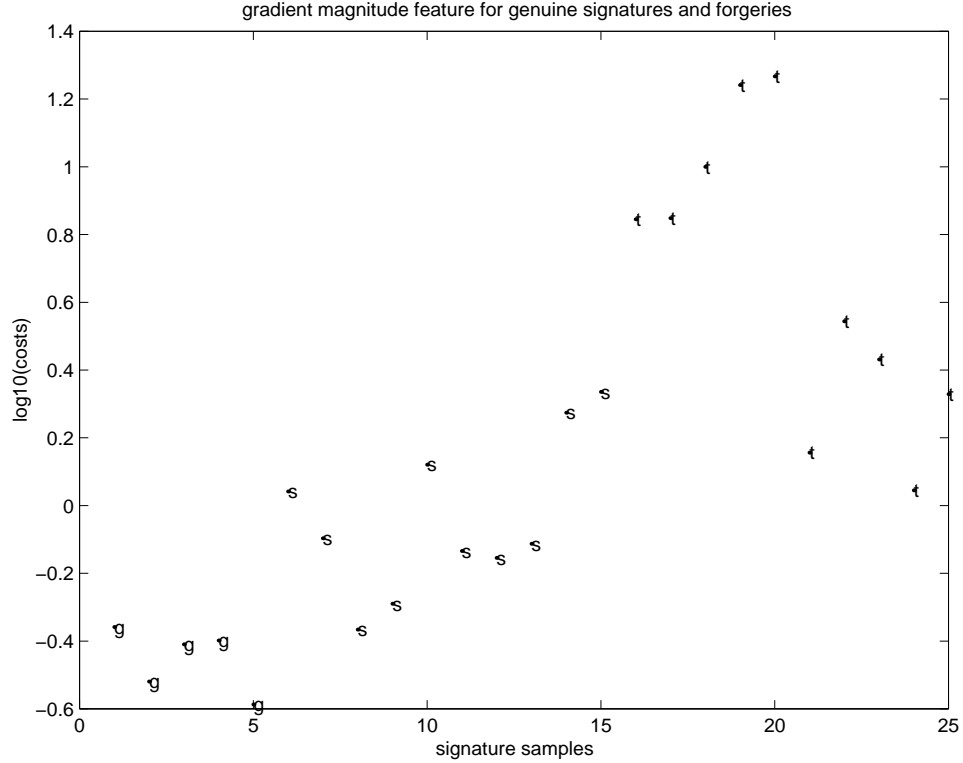
Figure 5.20: An example of the cost of gradient magnitude.

segment. The obtained edge profiles include the gradient magnitude and gradient direction along both edges. Using these values of gradient magnitude and gradient direction, we compare their average values over each stroke segment and compute the cost for each feature as the sum of the costs over all the segments. The stroke width is also calculated using the gradient direction information along each stroke segment. The costs for stroke width and gray level are calculated as described above.

In Figure 5.20 we show an example of the costs for gradient magnitude. In this Figure, the horizontal axis represents the signature samples, while the vertical axis shows the costs. The genuine signatures are labeled "g", the simulated forgeries are labeled "s" and the traced forgeries are labeled "t". Similar results for gradient direction are shown in Figure 5.21. The costs for gray level and stroke width are shown in Figure 5.22 and Figure 5.23. In the Figures, the costs are the logarithms of the costs computed as described above. Since some of the costs are considerably higher than others, taking the logarithm gives better plots. The first five samples are the genuine signatures, the next ten samples are simulated forgeries, and the last ten are traced forgeries. The costs do in fact tend to be lower for the genuine signatures. The total costs are plotted in Figure 5.24; they show a good separation between the genuine signatures and the forgeries.

Table 5.2 shows the the error rates obtained on our data set using each feature separately when we use hand-selected thresholds. We did not attempt to fit Gaussians to the data and find minimum-error thresholds, because there were too few samples to give meaningful fits.

As discussed in Section 5.3, the gradient should be a useful feature because it reflects the forces and angles of the writing instrument. Differences in how the writing instrument is held are common among different authors. The gray level also changes along the stroke. However,
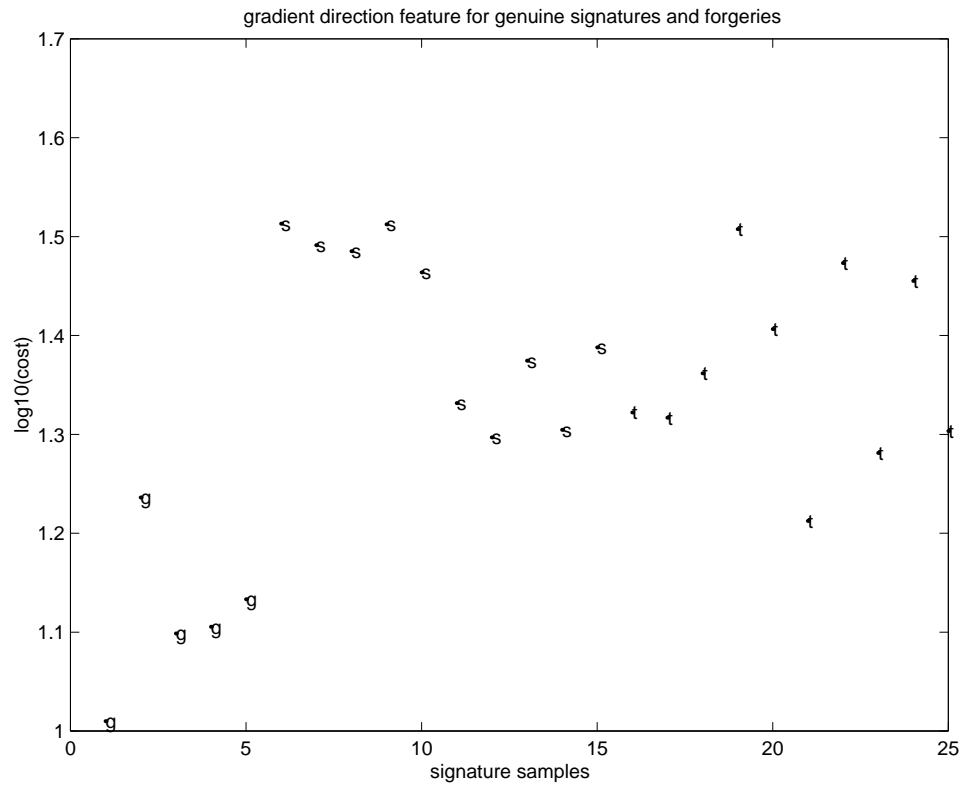
91

Figure 5.21: An example of the cost of gradient direction.

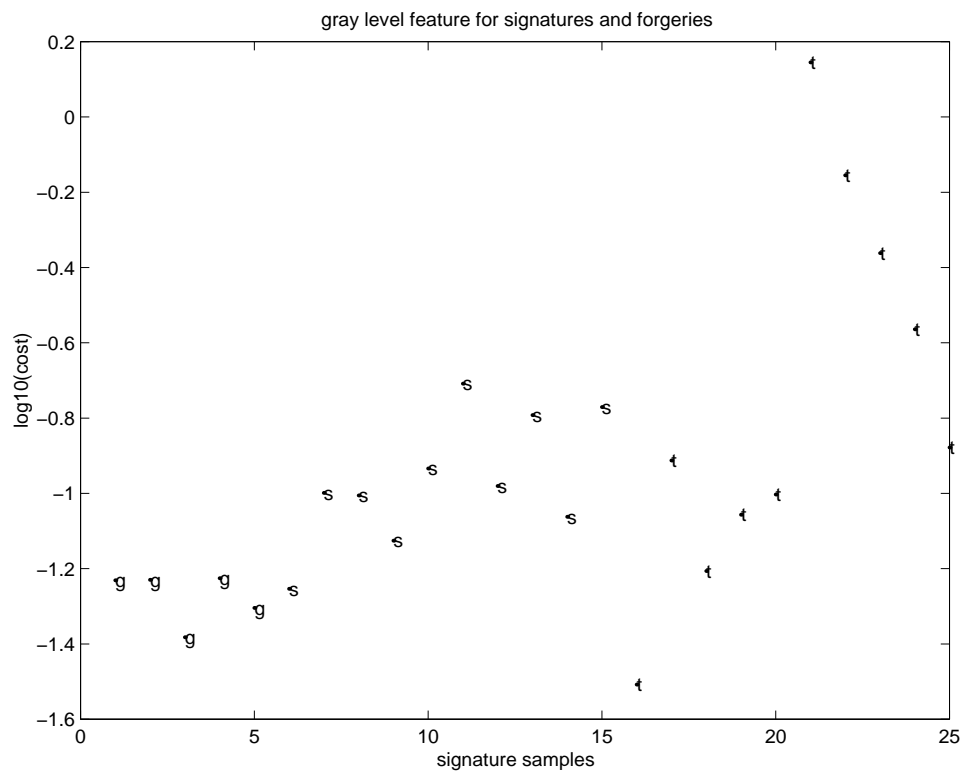| features used	type I error	type II error
gradient magnitude	7/50(14%)	30/200(15%)
gradient direction	6/50(12%)	34/200(17%)
gray level	9/50(18%)	44/200(22%)
width	10/50(20%)	35/200(17.5%)
sum over all features	3/50(6%)	23/200(11.5%)

Table 5.2: Error rates.

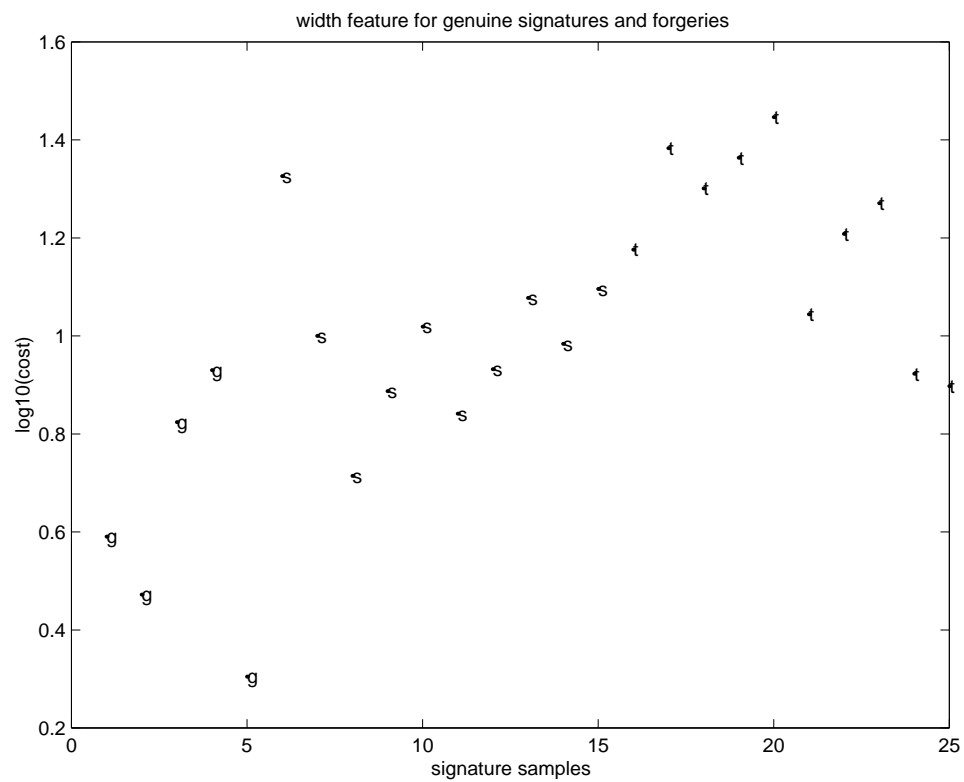Figure 5.22: An example of the cost of gray level.

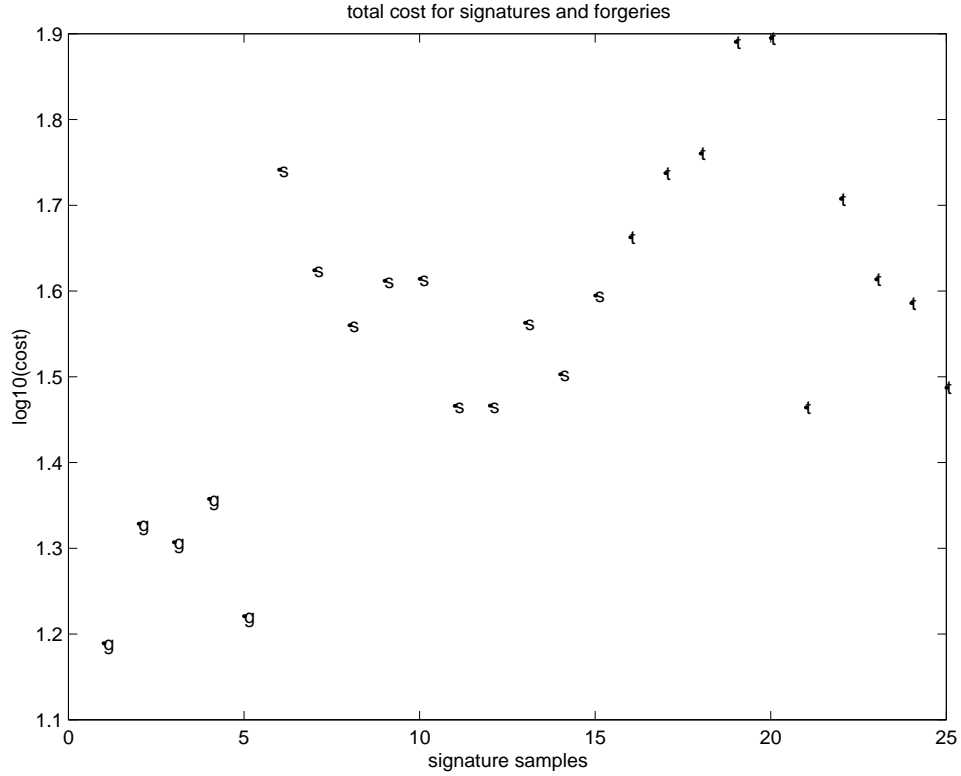Figure 5.23: An example of the cost of width.

Figure 5.24: Total cost of gradient magnitude, gradient direction, gray level and width.

due to the effects of noise and the automatic image enhancement performed in most commercial scanners, the changes in gray level are less obvious. Differences between writing instruments also make it difficult to detect the changes in gray level along a stroke, even the changes from one stroke segment to another. The width feature is also affected by the writing instrument, and it can also be hard to calculate. However, all four features contribute to the distinctiveness of a person's writing. Among them, which one is the dominant feature depends on the individual.

In most of the cases, the gray level cost for traced forgeries seems to be higher than that for freehand forgeries. Even though different people write different portions of a stroke, e.g., the upward and downward strokes of a loop, differently, they tend to press harder on the downward portion than the upward portion of the stroke when writing ballistically. When tracing a signature, however, the difference will not be as obvious as in natural writing. Similar arguments hold for gradient magnitude, and here too we find that the cost for traced forgeries is usually higher than the cost for freehand forgeries. The ballistic change along the edge and the difference between the two edges are determined by the force and angle change while writing. The combination of force and angle together with the fact that the cost is calculated along two sets of edges may make the gradient magnitude more distinguishable than the grey level, which is mainly determined by the force. Also, the gray level is only calculated from one trace. As a result, the gradient magnitude has an advantage over the gray level, as confirmed by our results.

5.4.3 Discussion of results

In summary, the production of a signature is affected by a combination of many factors. Two important ones are the force exerted on the writing instrument and the angle at which it is held. The grey level and stroke width both depend on the force and the angle, but they are different combinations of these factors. Which feature dominates the cost is thus writer-dependent.

As discussed above, traced forgeries may have high costs for gradient magnitude, gray level and stroke width, but the direction feature cost for traced forgeries is likely to be smaller than that for freehand forgeries. This is because a traced forgery closely follows a genuine signature, which determines the direction histogram of the stroke segment. Sometimes, tremor along a stroke segment can make the direction cost high for a traced forgery. Generally speaking, however, these effects are smoothed out by taking the direction histogram of the segment. Another observation is that traced forgeries have a bigger standard deviation of costs because tracing is almost never done ballistically, and drawing can be more inconsistent from instance to instance compared with writing.

The four features together have considerable distinguishing power. Since the four feature values in the cost are not in the same numerical ranges and the feature that works best for a given writer varies from person to person, it is not appropriate to just add the costs of the different features together. Intuitively, we should weight each feature by the standard deviation of that feature for the available genuine signatures. For a feature with a large standard deviation, the weight should be smaller compared to a feature with a small standard deviation. As we can see from Table 5.2, the combination of the four features works better than any of the features alone even when they are not weighted.

Another advantage of having a sub-stroke-wise correspondence between the model and the questioned signature is that it provides us with an opportunity to prioritize the strokes. Expert document examiners know not only which clues to look for, but also where to look, i.e., which strokes to pay more attention to. Different stroke segments play different roles in a signature. Putting aside personal touches in different stroke segments, a horizontal stroke may not mean as much as a vertical stroke as regards width changes, while a loop is definitely harder to forge compared with a straight line, either horizontal or vertical. If we can pick only stroke segments that are characteristic of each mode, or weight different stroke segments differently according to their discriminating power, we should be able to improve our results.

To illustrate the concept of using selected stroke segments for verification, we use the same examples as in Figures 5.20-5.24. In Figure 5.25 and Figure 5.26, about 1/3 of the stroke segments are selected to compute the costs of gradient magnitude and gradient direction. We see that when this is done, the difference between the average costs of genuine signatures and forgeries becomes larger. If we chose a different set of stroke segments, the results would be different. In Figure 5.25 and Figure 5.26, relatively long stroke segments were chosen.

Figure 5.27 shows an example of the direction feature for a single stroke segment. The stroke segment is shown in Figure 5.27(a); it is the one pointed to by the arrow. It is a relatively straight segment in the middle of a signature. Due to the nature of this segment, it may not have significant changes in grey level or width. However, since it is a connecting stroke segment between two neighboring letters, the angles at which it leaves the previous stroke segment and enters the next stroke segment are critical. The slant of such a stroke segment is hard to imitate. Thus the direction feature is dominant for this stroke segment for the freehand forgeries. For the traced forgeries, however, we would not expect this feature to be significant; the slant of the segment can be traced very well.

Figure 5.28 shows an example in which the gradient magnitude is an important feature. As
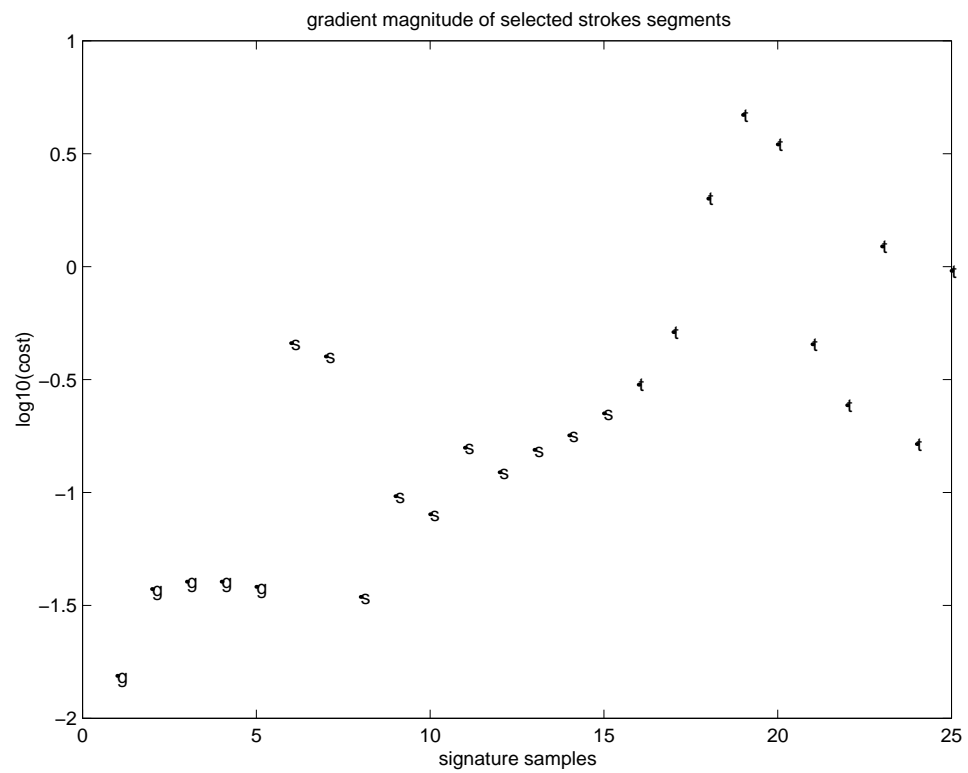
Figure 5.25: An example of the cost of gradient magnitude using a selected set of stroke segments.
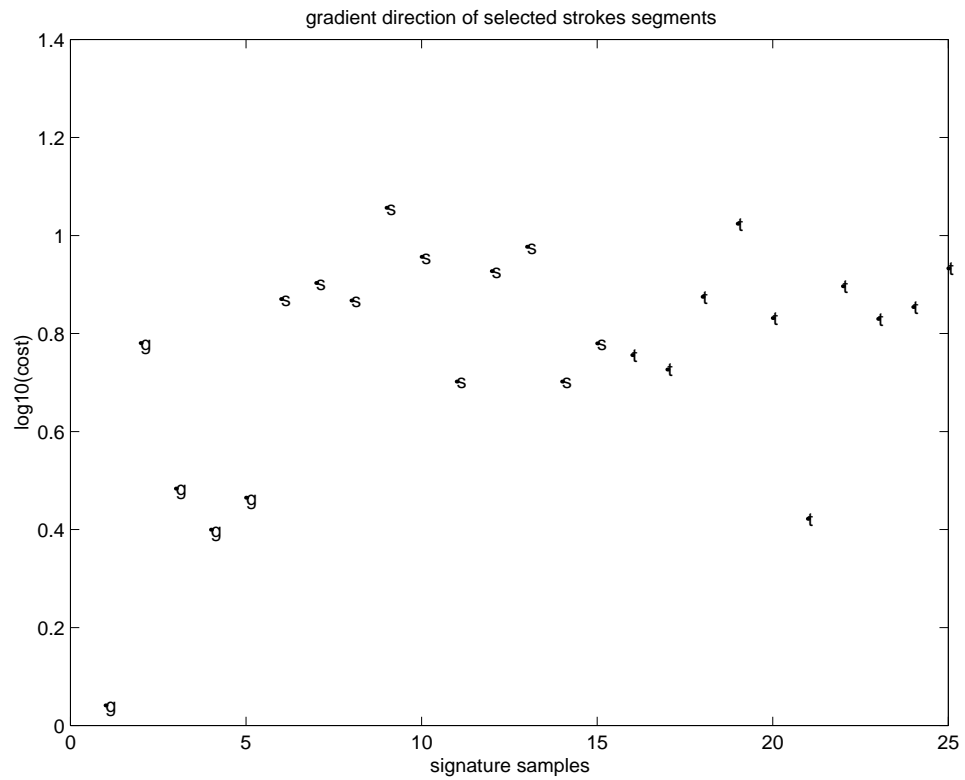
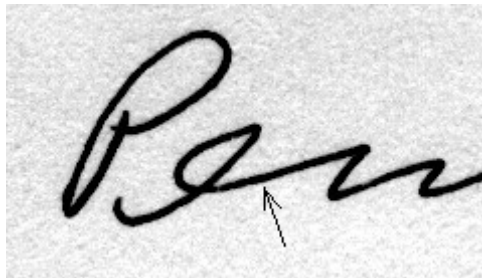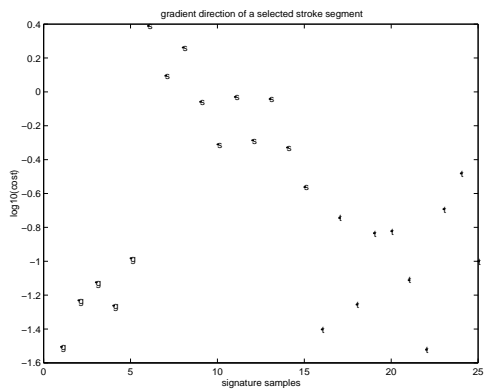Figure 5.26: An example of the cost of gradient direction using a selected set of stroke segments.

(a)   (b)

Figure 5.27: Example illustrating the direction feature: (a) The stroke segment. (b) The direction feature cost.
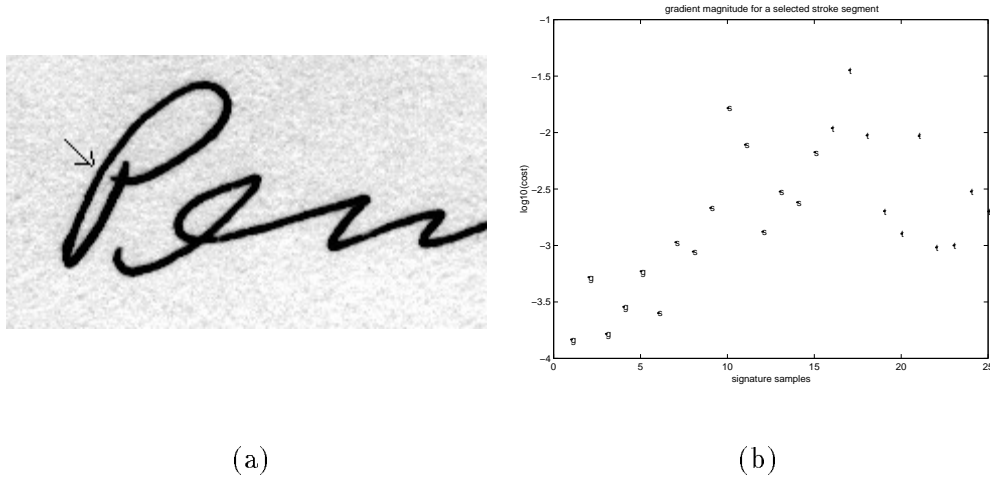
97

(a) (b)

Figure 5.28: Example illustrating the gradient magnitude feature: (a) The stroke segment. (b) The gradient magnitude feature cost.

shown in Figure 5.28(a), the stroke segment is a relatively long upstroke. For such a stroke, writing ballistically or non-ballistically makes a difference in the gradient magnitude. Unlike the gradient direction feature, the traced forgeries do not do well here since tracing involves more drawing and less ballistic writing.

Finally, Figures 5.29 and 5.30 show examples of the gray level and width features for a downstroke segment. When written ballistically, this type of stroke tends to be darker and wider than an upstroke. It also should be darkest and widest at the end of the segment, which may be either an end point or a turn point where the downstroke becomes an upstroke. For some people, the changes in darkness and width are more obvious for a downstroke than the changes in other type of stroke segments.

To further illustrate the advantage of selecting stroke segments for verification, Figure 5.31 shows an example involving two stroke segments. Figure 5.31(a) shows the segments, labeled "a" and "b". Their costs are shown in Figures 5.31(b) and (c). The first segment is a downstroke just after the connecting stroke that we showed in Figure 5.27(a), while the second segment is a small and more horizontally oriented. In this example, segment "a" has a better distinguishing power than segment "b".

5.5 Conclusion

Making a correspondence between a model and a questioned signature enables us to do a local comparison between them on a stroke or sub-stroke level. In this chapter, we describe the use of local features to detect skilled forgeries. The features studied are gradient magnitude, gradient direction, gray level, and stroke width. These features directly relate to the position and angle at which people hold a pen while writing. Even though the writing instrument and scanning device can influence the effectiveness of these features, they still provide information about personal characteristics that can be captured.

We have shown that these features can be used for skilled forgery detection. For different writers, which feature is dominant can be different. Future research could involve introducing

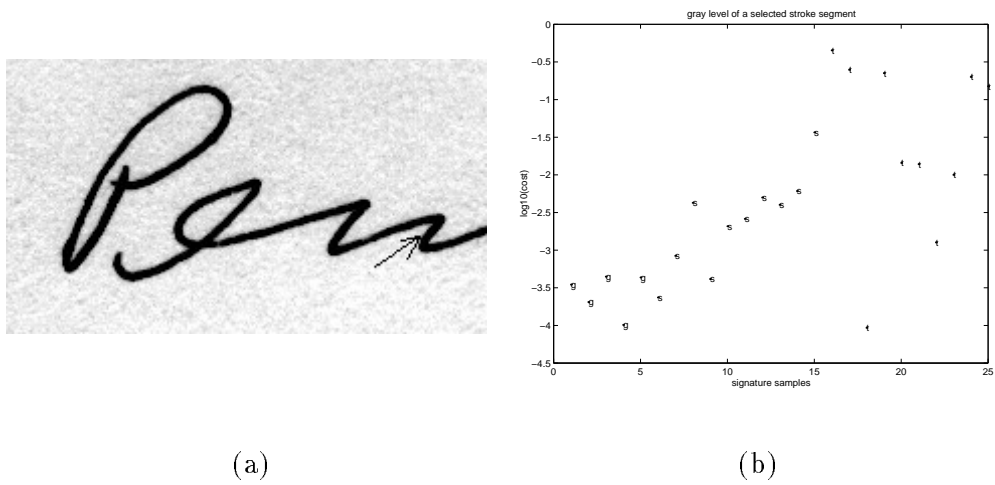Figure 5.29: Example illustrating the gray level feature: (a) The stroke segment. (b) The gray level feature cost.

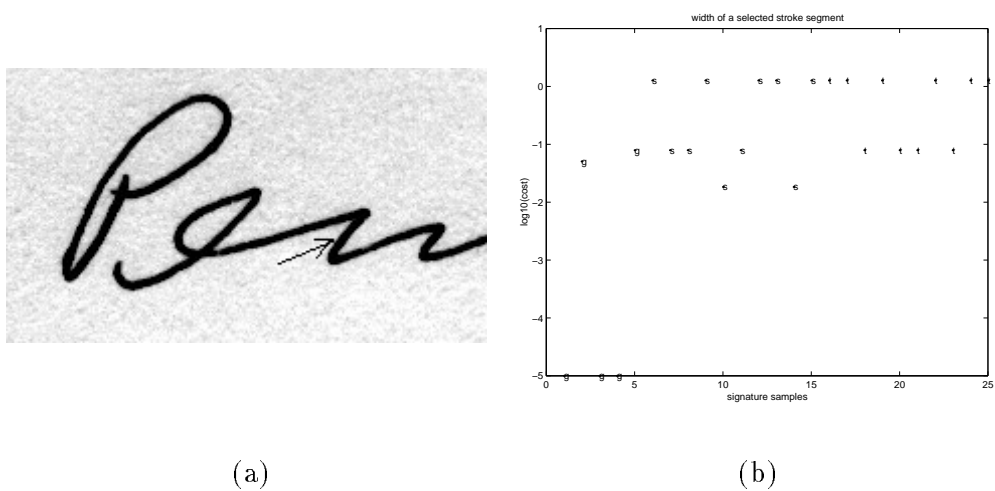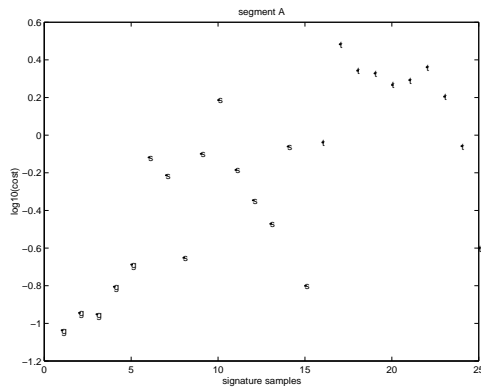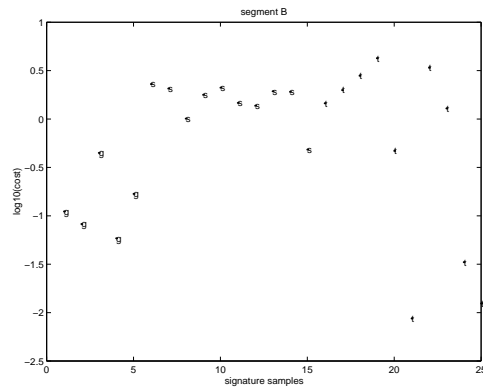
Figure 5.30: Example illustrating the width feature: (a) The stroke segment. (b) The width feature cost.

(a)

(b)   (c)

Figure 5.31: Comparison of a pair of stroke segments: (a) The segments. (b) The costs for segment "a". (b) The costs for segment "b".

weights for the different features for each writer. To train and test these weights, more data would be needed. Another topic for future research is how to more effectively measure the features, especially how to eliminate the effects of noise on the gradient and the grey level.

Unlike random or simple forgery detection, which may be used in places like check clearing houses for mass verification, skilled forgery detection aims at helping expert document examiners in processing a small volume of data. Thus, it is possible that we can let the document examiner pick out the candidate stroke segments to be studied. These candidate segments can be generic stroke segments which show greater feature value differences overall, like upward and downward strokes, or they can be any stroke segments that have special personal touches. The overall cost will then be calculated based on the selected stroke segments. If an expert could choose stroke segments for comparison, the results that we illustrated in Figures 23 and 24 would undoubtedly be improved further.

Chapter 6

Conclusions and Further Research

6.1 Contributions of the Dissertation

In this thesis we have addressed the problem of signature verification. We discussed the nature of signatures and forgeries and reviewed the related literature. We showed that as signature databases grow larger and researchers move towards more difficult forgery detection tasks, especially skilled forgery detection, more local features needed to be examined in addition to global features.

We have presented algorithms for making a local correspondence between a model and a questioned signature. In Chapter 3, we presented an algorithm that makes such a correspondence on the sub-stroke level. In Chapter 4, we presented an algorithm that makes a point-wise correspondence. Both algorithms enable us to examine questioned signatures on a stroke level. Results on random and simple forgery detection are presented. The main difference between the two algorithms is in how the signatures are segmented into meaningful stroke segments. One algorithm uses the edge information without prior model guidance; the other uses model information. When making the correspondence, the model was used in the first algorithm to guide the search for consecutive stroke segments. The correspondence between the model and the questioned signature not only enables us to eliminate random and simple forgeries, but also provides us with a tool for examining the local properties needed for skilled forgery detection.

Another contribution of this thesis is that it introduced the idea of weighting different features for different stroke segments. Expert document examiners never treat all strokes the same when they examine a questioned document. In addition, each person has certain significant features in his/her writing. These features should be the ones that a verification system focuses on. By using the reciprocal of the standard deviation as a weighting factor, we emphasize the less varying features of a stroke segment instead of treating all the features of all the stroke segments equally.

We also make use of the idea of fitting a Gaussian function to the cost distribution and using the Gaussian parameters to calculate a threshold for discriminating between signatures and forgeries. This provides a tool for automatically selecting thresholds. In our experiments, the thresholds used for signature verification were chosen theoretically rather than empirically; they were based on statistical models, and were not hand-picked. This gives us some idea of how our algorithm would perform if we added more signatures to the database, provided the statistical models are valid.

One thing that needs to be pointed out is that we arbitrarily selected the signatures for the reference set. In our experiment, we selected the first (fixed number of) signatures for training the weights as well as training the thresholds. There has been research on optimizing the selection of reference sets for both on-line and off-line signature verification. In [72], Allgrove

and Fairhurst developed methods for maximizing the performance of signature verification systems. Similar research has been performed for on-line signature verification systems [73]. The model in our experiment was generated from the first signature of each person in the database without careful selection.

In our experiments, we used only a single model for each writer. If we generated multiple models for each writer and tested the questioned signature against each model, we should be able to achieve better results. We could use a majority vote on the cost functions, or use a Bayesian classifier to choose among the cost functions corresponding to the different models.

Finally, we presented a skilled forgery detection algorithm that made use of writer-dependent information embedded at the sub-stroke level. This was made possible by the correspondence between the model and the questioned signature. The difference in our approach as compared with others is that we tried to capture the unballistic and tremor information in each stroke segment instead of as global statistics. We also briefly explored the idea of selectively choosing the stroke segments for skilled forgery detection. As mentioned earlier, document examiners look only at strokes that are significant to each writer. If we can develop a system in which their trained eyes provide the stroke segments to be examined, while we provide them with quantitative ways of evaluating the signatures, it will aid in the processing of signatures in forensic science.

6.2 Future Extensions

Our matching method worked well in our experiments, but a great deal of work will have to be done to develop a system based on our method that would be usable in a real environment.

In Chapter 3 and Chapter 4 we used different stroke segmentation schemes. They both have pros and cons. The edge-based method relies on the quality of the edge image and needs another step to get the correspondence with the model. The model-guided approach, on the other hand, made good use of the model information to guide the processes of segmentation and correspondence. However, it may not have as much informative power as the first approach. If we can combine the two approaches, we can use the model to guide the process only at junction points where without the model, it is hard to decide where the stroke continues. The model can also be used when the edge of the stroke breaks due to noise. If we do this, we also do not have to confront the global vs. local optimization issue raised when we limit the depth of the tree. Another advantage of combining the two approaches is that we don't have to consider the issue of choosing the first points on the questioned signatures.

As mentioned above, our model was generated from the first signature in each writer's database. Even though people sign their names consistently, non-typical instances occur once in a while. If we happen to choose an instance that is not consistent with the writer's usual signature, this will no doubt affect performance. We should be able to quantitatively study the reference signatures and generate the model from the one that is most typical for that writer.

The same argument applies to the reference signatures from which the weights and thresholds are generated. In one experiment [72], it was in fact shown that removal of the non-validated signature models can significantly improve error rates in a signature verification system.

Another issue that needs further study is how to measure the features more robustly. As we have already seen in Chapter 5, digitized noise can be a problem for feature extraction. It not only causes high-frequency components as we have seen, but also causes the edges to break making it is difficult to estimate width as well as gradient information.

The idea used in our approach can also be adopted in related areas. One possible application for the local correspondence between the model and the questioned signature is in Optical

Character Recognition. If we can recover the temporal information involved in writing the letters and characters, it can lead the way to recognizing them. Chapter 4 showed some examples involving single letters.

A problem similar to signature verification is writer identification. As pointed out by expert document examiners, writing style develops and matures in each individual, especially the special touches that people use in various letters and in positions of the words. The idea of weighting different features in different people's writing differently can be adopted in treating the problem of writer identification.

Bibliography

[1] F. Leclerc and R. Plamondon. Automatic signature verification: The state of the art — 1989-1993. *International Journal of Pattern Recognition and Artificial Intelligence*, 8:3–20, 1994.

[2] R. Plamondon and G. Lorette. Automatic signature verification and writer identification — State of the art. *Pattern Recognition*, 22:107–131, 1989.

[3] R. Plamondon and S.N. Srihari. On-line and off-line handwriting recognition: A comprehensive survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:63–84, 2000.

[4] R. Saferstein. *Forensic Science Handbook*. Prentice-Hall, 1982.

[5] O. Hilton. A study of the influence of alcohol on handwriting. *Journal of Forensic Sciences*, 14:308–316, 1969.

[6] O. Hilton. Consideration of the writer's health in identifying signatures and detecting forgery. *Journal of Forensic Sciences*, 14:157–166, 1969.

[7] A.S. Osborn. *Questioned Documents*. Patterson Smith Publishing Corporation, 1973.

[8] J. F. Masson. A study of the handwriting of adolescents. *Journal of Forensic Sciences*, 33:167–175, 1988.

[9] J.V.P. Conway. *Evidential Documents, Their Scientific Examination*. Springfield, 1959.

[10] E.H.W. Schroeder. A classification system for fraudulent checks. *Journal of Forensic Sciences*, 16:162–175, 1971.

[11] O. Hilton. Line quality - historic and contemporary views. *Journal of Forensic Sciences*, 32:118–120, 1987.

[12] S.C. Leung, Y.S. Cheng, H.T. Fung, and N.L. Poon. Forgery I – simulation. *Journal of Forensic Sciences*, 38:402–412, 1993.

[13] S.C. Leung, Y.S. Cheng, H.T. Fung, and N.L. Poon. Forgery II – tracing. *Journal of Forensic Sciences*, 38:413–424, 1993.

[14] D. Doermann. *Document Image Understanding: Integrating Recovery and Interpretation*. PhD thesis, University of Maryland at College Park, 1993.

[15] J. C. Pan and S. Lee. Offline tracing and representation of signatures. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 679–680, 1991.

[16] G. Dimauro, S. Impedovo, and G. Pirlo. Component-oriented algorithms for signature verification. *International Journal of Pattern Recognition and Artificial Intelligence*, 8:131–153, 1994.

[17] C.C.Tappert. Cursive script recognition by elastic matching. *IBM Journal of Research and Development*, 26:765–771, 1982.

[18] Y.Fujimoto, S.Kadota, S.Hayashi, M.Yamamoto, S.Yajima, and M.Yasuda. Recognition of handprinted characters by nonlinear elastic matching. In *Processing, International Conference on Pattern Recognition*, pages 113–118, 1982.

[19] M.J. Revillet. Signature verification on postal cheques. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 767–773, 1991.

[20] W.F.Nemcek and W.C.Lin. Experimental investigation of automatic signature verification. *IEEE Transactions on Systems, Man and Cybernetics*, 4:121–126, 1974.

[21] P.C.Chuang. machine verification of handwritten signature image. In *Proceedings, International Conference on Crime Countermeasures*, pages 105–109, 1977.

[22] B. Zhang, M. Fu, and H. Yan. Handwritten signature verification based on neural 'gas' based vector quantiztion. In *Proceedings of the Fourteenth International Conference on Pattern Recognition*, pages 1862–1864, 1998.

[23] Y. Mizukami, H. Miike, M. Yoshimura, and I. Yoshimura. An off-line signature verification system using an extracted displacement function. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 757–760, 1999.

[24] R.N. Nagel and A. Rosenfeld. Computer detection of freehand forgeries. *IEEE Transactions on Computers*, 26:895–905, 1977.

[25] M. Ammar, Y. Yoshida, and T. Fukumura. A new effective approach for off-line verification of signatures by using pressure features. In *Proceedings, International Conference on Pattern Recognition*, pages 566–569, 1986.

[26] R. Sabourin and R. Plamondon. Preprocessing of handwritten signatures from image gradient analysis. In *Proceedings, International Conference on Pattern Recognition*, pages 576–579, 1986.

[27] R. Sabourin and J.P. Drouhard. Off-line signature verification using directional PDF and neural networks. In *Proceedings, International Conference on Pattern Recognition*, volume 2, pages 321–325, 1992.

[28] J.P. Drouhard, R. Sabourin, and M. Godbout. A comparative study of the k-neareast neighbour, threshold and neural network classifiers for handwritten signature verification using an enhanced directional PDF. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 807–810, 1992.

[29] J.P. Drouhard, R. Sabourin, and M. Godbout. A neural network approach to off-line signature verification using directional PDF. *Pattern Recognition*, 29:415–424, 1996.

[30] H. Cardot and M. Revenu. A static signature verification system based on a cooperating neural networks architecture. *International Journal of Pattern Recognition and Artificial Intelligence*, 8:39–52, 1994.

[31] N.A.Murshed, F.Bortolozzi, and R. Sabourin. Off-line signature verification, without a priori knowledge of class ω_2. a new approach. In *Proceedings of the International Conference on Document Analysis and Recognition*, 1995.

[32] M. Dehghan, K. Faez, and M. Fathi. Signature verification using shape descriptors and multiple neural networks. In *Proceeding of IEEE Region 10 Annual Conference. Speech and Image Technologies for Computing and Telecommunication*, pages 415–418, 1997.

[33] N. Papamarkos and H. Baltzakis. Off-line signature verification using multiple neural network classification. In *Proceeding of 13th International Conference on Digital Signal Processing*, pages 727–730, 1997.

[34] X. Xiao and G. Leedham. Signature verification by neural networks with selective attention. *Applied Intelligence*, 11:213–223, 1999.

[35] R. Sabourin, M. Cheriet, and G. Genest. An extended-shadow-code-based approach for off-line signature verification. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 1–5, 1993.

[36] R. Sabourin and G. Genest. An extended-shadow-code-based approach for off-line signature verification: Part I. Evaluation of the bar mask definition. In *Proceedings, International Conference on Pattern Recognition*, volume 2, pages 450–453, 1994.

[37] R. Sabourin and G. Genest. An extended-shadow-code-based approach for off-line signature verification: Part II. Evaluation of several multi-classifier combination strategies. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 197–201, 1995.

[38] C. Simon, E. Levrat, R. Sabourin, and J. Bremont. A fuzzy perceptron for off-line handwritten signature verification. In *Proceedings of the Brazilian Symposium on Document Image Analysis*, pages 261–272, 1997.

[39] R. Sabourin, G. Genest, and F. Preteux. Pattern spectrum as a local shape factor for off-line signature verification. In *Proceedings, International Conference on Pattern Recognition*, volume 3, pages 43–48, 1996.

[40] R. Sabourin, G. Genest, and F. Preteux. Off-line signature verification by local granulometric size distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:976–988, 1997.

[41] R. Sabourin, J.P. Drouhard, and E.S. Wah. Shape matrices as a mixed shape factor for off-line signature verification. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 661–664, 1997.

[42] F. Nouboud and R. Plamondon. Global parameters and curves for off-line signature verification. In *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pages 145–155, 1994.

[43] L.L. Lee and M.G. Lizarraga. An off-line method for human signature verification. In *Proceedings, International Conference on Pattern Recognition*, volume 3, pages 195–198, 1996.

[44] T. Pavlidis, R. Mavuduru, and N. Papanikolopoulos. Off-line recognition of signatures using revolving active deformable models. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pages 771–776, 1994.

[45] T. Pavlidis and N.P. Papanikolopoulos. Application of deformable structures to signature identification. In *Proceedings of the International Conference on Digital Signal Processing*, pages 578–583, 1995.

[46] K. Han and K. Sethi. Signature identification via local association of features. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 187–190, 1995.

[47] G. Wilensky and R. Crawford. Recognition and characterization of handwritten words. In *Proceedings of the Symposium on Document Image Understanding Technology*, pages 87–98, 1997.

[48] J. Guo, D. Doermann, and A. Rosenfeld. Local correspondence for detecting random forgeries. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 319–323, 1997.

[49] R. Baron and R. Plamondon. Acceleration measurement with an instrumented pen for signature verification and handwriting analysis. *IEEE Transactions on Instrumentation and Measurement*, 38:1132–1138, 1989.

[50] R. Plamondon and M. Parizeau. Signature verification from position, velocity and acceleration signals: A comparative study. In *Proceedings, International Conference on Pattern Recognition*, pages 260–265, 1988.

[51] J.J. Brault and R. Plamondon. Histogram classifier for characterization of handwritten signature dynamic. In *Proceedings, International Conference on Pattern Recognition*, pages 619–622, 1984.

[52] G. Dimauro, S. Impedovo, and G. Pirlo. A signature verification system based on dynamical segmentation technique. In *Proceedings of the International Workshop on Frontiers in Handwriting Recognition*, pages 262–271, 1993.

[53] R. Plamondon. A model-based segmentation framework for computer processing of handwriting. In *Proceedings, International Conference on Pattern Recognition*, pages 303–307, 1992.

[54] Y. Sato and K. Kogure. On-line signature verification based on shape, motion and handwriting pressure. In *Proceedings, International Conference on Pattern Recognition*, pages 823–826, 1982.

[55] M. Parizeau and R. Plamondon. A comparative analysis of regional correlation, dynamic time warping and skeletal tree matching for signature verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:710–717, 1990.

[56] W. Lee, N. Mohankrishnan, and M.J. Paulik. Improved segmentation through dynamic time warpint for signature verification using a neural network classifier. In *Proceedings of the 1998 International Conference on Image Processing*, pages 929–933, 1998.

[57] N.M.Herbst and C.N.Liu. Automatic signature verification. In *Computer Analysis and Perception*, pages 83–105. CRC Press, 1982.

[58] L. Yang, B.K.Widjaja, and R.Prasad. On-line signature verification applying hiden markov models. In *Proceedings, Scandinavian Conference on Image Analysis*, pages 1311–1316, 1993.

[59] R. Kashi, J. Hu, W. Nelson, and W. Turin. A hidden Markov model approach to on-line handwritten signature verification. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 253–257, 1997.

[60] J.G.A. Dolfing, E.H.L. Aarts, and J.J.G.M. van OOsterhout. On-line signature verification with hidden markov models. In *Proceedings of the Fourteenth International Conference on Pattern Recognition*, pages 1309–1312, 1998.

[61] G. Rigoll and A. Kosmala. A systematic comparison between on-line and off-line methods for signature verification with hidden markov models. In *Proceedings of the Fourteenth International Conference on Pattern REcognition*, pages 1755–1757, 1998.

[62] K.Kiriyama H.Taguchi, E.Tanaka, and K.Fujii. On-line recognition of handwritten signatures by feature extraction of pen movements. *System computers Japan*, 20:1–14, 1989.

[63] M. Munich and P. Perona. Camera-based ID verification by signature tracking. In *Proceedings, European Conference on Computer Vision*, 1998.

[64] M.E. Munich and P. Perona. Visual signature verification using affine arc-length. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 180–186, 1999.

[65] K.K.Paliwal, A.Agrawal, and S.S.Sinha. Modification of Sakoe and Chiba's dynamic warping algorithm for isolated word recognition. In *IEEE Proceedings, International Conference on Acoustics, Speech and Signal Processing*, pages 1259–1261, 1982.

[66] M.Yoshimura and I.Yoshimura. An application of the sequential dynamic programming matching method to off-line signature verification. In *Advances in Document Image Analysis*, pages 299–310. Springer, 1997.

[67] M. Ammar. Progress in verification of skillfully simulated handwritten signatures. *International Journal of Pattern Recognition and Artificial Intelligence*, 5:337–351, 1991.

[68] B. Fang, Y.Y. Wang, C.H. Leung, Y.Y. Tang, P.C.K. Kwok, K.W. Tse, and Y.K.Wong. A smoothness index based approach for off-line signature verification. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 785–787, 1999.

[69] M. Worring and A. W. M. Smeulders. Digital curvature estimation. *CVGIP: Image Understanding*, 58:366–382, 1993.

[70] J. F. Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8:679–698, 1986.

[71] S.W. Zucker, R.A. Hummel, and A. Rosenfeld. An application of relaxation labeling to line and curve enhancement. *IEEE Transactions on Computers*, 26:922–929, 1977.

[72] C.C. Allgrove and M.C. Fairhurst. Optimisation issues in dynamic and static signature verification. In *IEE Third European Workshop on Handwriting Analysis and Recognition*, pages 5/1–5/6, 1998.

[73] V. Di Lecce, G. Dimauro, A. Guerriero, S. Impedovo, G. Pirlo, A. Salzo, and L. Sarcinella. Selection of reference signatures for automatic signature verification. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 597–600, 1999.